The Dynamics of Reasoning

Engelfriet, J.

1999

The Dynamics of Reasoning

VRIJE UNIVERSITEIT

The Dynamics of Reasoning

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit te Amsterdam,
op gezag van de rector magnificus
prof.dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
wiskunde en informatica
van de faculteit der exacte wetenschappen
op donderdag 4 februari 1999 om 15.45 uur
in het hoofdgebouw van de universiteit,
De Boelelaan 1105

door

JOHANNES ENGELFRIET

geboren te Leiden

Promotor: prof.dr. J. Treur

Acknowledgments

First and foremost, I would like to thank my thesis advisor and promotor, Jan Treur. He has always been an enthusiastic and encouraging mentor in the world of science, and the cooperation with him has been inspiring and pleasant. Furthermore, I would like to thank my reading committee (Johan van Benthem, Michael Fisher, Jan-Willem Klop, John-Jules Meyer and Rineke Verbrugge), for reading and commenting on an earlier version of this thesis, in particular Rineke Verbrugge who read more than one version and whose many comments have much improved the result. I would like to express my gratitude to Johan van Benthem and Dick de Jongh, who supervised my master's thesis for mathematics, which lay the foundation for part of the work reported here.

Of course I owe much to the rest of the scientific environment around me: the Department of Artificial Intelligence and the rest of the staff at the Faculty of Sciences (Mathematics and Computer Science before that), the Dutch AI and Logic colleagues (and the Dutch Research School in Logic — Onderzoeksschool Logica) and the international AI and Logic community which I was fortunate to meet at workshops and conferences. Especially I thank all my co-authors: Frances Brazier, Pascal van Eck, Dieter Fensel, Frank van Harmelen, Heinrich Herre, Catholijn Jonker, Wiktor Marek, Jan Treur, Mirek Truszczyński, Yde Venema, and Mark Willems.

During my time as a PhD student, I have been able to make many travels thanks to the financial support of the Faculty of Mathematics and Computer Science, NWO, Shell Nederland B.V. and the ESPRIT Basic Research project 6156 DRUMS II.

Last but not least I would like to thank everyone who gave me moral support: my parents (Louco and Joost Engelfriet), friends, family, and of course especially my partner, Elles de Vries.

Contents

Chapter 1

Introduction

This thesis is about reasoning processes that occur in practice, for instance when a diagnostic expert performs his or her task. We shall start with an (informal) discussion of reasoning processes.

1.1 Reasoning processes

In a very broad sense, reasoning can be seen as an activity of an agent, resulting in a change of the agent's mental state. This mental state at least captures the agent's informational attitude besides other mental attitudes considered in the agent literature, e.g., motivational attitudes such as desires, intentions and plans (see [WJ95] or [Bra87]). This thesis focuses on informational attitudes, leaving motivational attitudes untouched. Such an informational attitude may consist of, for instance, knowledge, beliefs, or assumptions the agent has. Although the status of these kinds of information is different, we will consider the abstract features relevant to all of them. In this thesis, we will sometimes use the terms 'knowledge' ('knows') or 'belief' ('believes'), but only as synonyms for 'information', so these terms do not impose a special status on the information of the agent. We will use the word 'agent' only in the sense of 'reasoner' or 'reasoning entity', so we will not assume an agent has any special properties (such as pro-activeness or social ability, often assumed in the multi-agent literature; see for example [WJ95]).

We will call the part of the agent's mental state capturing information, an *information state*. An important assumption we will make, is that reasoning is a discrete process: starting from an (initial) information state, the agent invokes some kind of reasoning mechanism to arrive at a new information state, from which it again may perform some reasoning, possibly ad infinitum.

The study of classical modes of reasoning is at least as old as the syllogisms of Aristotle, and has led to various systems such as natural deduction. The reasoning we will consider in this thesis is allowed to involve phenomena of a more complex nature such as reasoning with defaults (plausible assumptions), revision, interaction (with the environment, through observations or communication), and introspection (meta-level reasoning). In general, both the reasoning path followed and the outcome of such reasoning processes may depend on decisions dynamically taken during the reasoning process.

Consider an agent that wants to buy a ticket for the film Rocky XIV. It has knowledge about movies in general, and about making reservations. But it also knows that in general, reservations are unnecessary. So by default, it decides not to reserve a seat. Then, however, it hears from a friend agent that Rocky films are always very popular. So, it decides to find out how long the film has been playing. To this end, it consults a newspaper and finds out that it is the first week. Therefore, it decides to make a reservation anyway.

In this scenario, we see an agent performing many forms of reasoning in an integrated fashion. It does classical reasoning, it performs default reasoning, it reasons about observations, it communicates, and it revises its knowledge on the basis of observation results and communicated information. The decision to perform an observation (looking in the paper) not only changed the reasoning path, but also changed the final outcome of the reasoning process. These are the kinds of practical reasoning, where the (internal) dynamic behavior of the agent may both influence the reasoning path and its final conclusions, that are the subject of the work reported here. The aim of this thesis is to present a general framework in which reasoning processes are formalized semantically (in an abstract way), and can be specified and studied.

1.2 Formalizing reasoning processes

In the field of nonmonotonic reasoning, many particular formalisms have been defined and studied, such as default logic (see [Rei80b]), autoepistemic logic (see [Moo85]), circumscription (see [McC80]), and nonmonotonic logic I and II (see [MD80] and [MD82]). After it became clear that none of these (or other) approaches would be *the* perfect account of commonsense reasoning, people started investigating general abstract properties of nonmonotonic reasoning. This line of research, started with a paper by Gabbay ([Gab85]), considers inference relations. The idea is that any form of nonmonotonic reasoning gives rise to a relation $\mid\!\sim$, called an inference relation, between (sets of) formulae and formulae, where $A \mid\!\sim \varphi$ means that φ can be concluded from the premises in A by using this form of reasoning. Each of the formalisms mentioned above indeed defines an inference relation (for most of them there are actually a number of variants). By abstracting from the particular formalisms defining such inference relations, general properties of nonmonotonic reasoning can be and actually have been studied. It proved to be a fruitful perspective, leading to the hallmark paper by Kraus, Lehmann and Magidor ([KLM90]); the inference relations of that paper (relations between formulae) were generalized to consequence operations (functions from potentially infinite sets of formulae to potentially infinite sets of formulae) and studied in [Mak94]. Many interesting phenomena of (nonmonotonic) reasoning can be (and have been) studied in the framework of inference relations and consequence operations. However, this framework only considers (purely functional) input–output behavior of reasoning. It focuses on the final product of reasoning: *what* can be derived in the end? In this thesis, we will also be interested in *how* these final conclusions are reached. In particular, two aspects of reasoning will be focussed on: nondeterminism and dynamics, to be explained below. In line with the study of consequence relations, we will investigate these aspects in an abstract sense, only on a lower (or more detailed) level of abstraction (see Section 1.3).

Nondeterminism

In formalization of reasoning using consequence operations, the focus is on the functional input–output behavior of a reasoning agent: it starts with a certain initial set of beliefs, and after reasoning it has a unique (usually) different set of beliefs. What actually happens in many forms of non-classical reasoning processes, is that there may be several possible sets of conclusions. This phenomenon occurs in formalizations of nonmonotonic reasoning (where these sets of conclusions are called *extensions* or *expansions*), in belief revision (removing a formula from a belief set in a minimal fashion is possible in different ways), and in the selection of observations to make. To arrive at a single set of conclusions, the agent must perform some operation on the set of possible sets of conclusions (like selecting one, or intersecting them). Such a reduction step entails a loss of information. Some people have an ambivalent or even negative attitude towards multiple extensions (some feel that the existence of multiple extensions shows that the reasoning formalism is wrong). However, in practical reasoning it is often the case that decisions taken during the reasoning process may lead to a different set of conclusions. This happens when (default) assumptions are made (and we have to choose which assumptions to make), when communicating with others (do we believe what the other person tells us?), when we get new information contradicting earlier beliefs or assumptions (and we have to decide which of our beliefs to give up). As this nondeterministic behavior is ubiquitous in complex reasoning, we want to study it in its own right.

Dynamics

The second aspect of practical reasoning that has, in our opinion, received too little attention in the past, is the importance and influence of the dynamics of the reasoning process. Reasoning is a process performed by an agent, taking time. Usually, the agent starts with some initial facts, then applies some rules (or another mechanism) to arrive at a different information state, from which it may again deduce conclusions. During this process, the agent goes through a number of possible information states; the process may even never end. The agent may also make decisions regarding its own reasoning process. Such meta-level reasoning may result in different goals, in the decision to apply a default or to perform an observation.

This kind of *control knowledge* has not been taken very seriously yet in the abstract study of reasoning. We share the view also put forward in [Ben91a] that integrating (dynamic) process aspects into the semantics of logical systems is more transparent and fruitful than trying to abstract from them (this is put into practice in e.g. [Eth87] but also implicitly in [Gab82]). This dynamic perspective fits into the recent general trend in logic towards studying the dynamics of information (witnessed by e.g., [Ben96a]), where we have a special interest in nonmonotonic reasoning from the standpoint of Artificial Intelligence. Studying the dynamics of reasoning allows us to investigate properties other than those referring just to input and output of the reasoning: does the reasoning end; is a conclusion drawn soon; how are conclusions drawn? Furthermore, making process aspects explicit is a first step towards resource-bounded reasoning, in which not only the mental capacities (like memory) of the agent are limited, but there is also limited time. Given time constraints, the agent may for example decide to focus on particular aspects of the domain.

We will take these two aspects, nondeterminism and reasoning dynamics, seriously, and we will analyze and formalize them both semantically and syntactically. On the one hand, this is inspired by an interest in these phenomena (occuring in practical reasoning) in their own right. By providing a formalization and specification framework of practical reasoning that takes into account nondeterminism and dynamics, we are able to study and compare different forms of reasoning, using mathematical and logical tools. This may have an influence on the static view of reasoning as well. Static properties of a certain form of reasoning can be seen to be 'caused', in a sense, by a (more refined) property of the nondeterministic view, or by a dynamic property. So the study of nondeterminism and dynamic aspects may further our understanding of the static view on reasoning. On the other hand, if we want to model, specify, study and reason about agents in dynamic environments, where the world the agent is reasoning about is changing, where information from other agents may come in at different points in time, where decisions have to be made in time, etcetera, a nondeterministic and dynamic view of reasoning will be *required*. A purely static view, if at all possible, will be insufficient to deal with all of these phenomena. Most of the material in this thesis will deal with the reasoning of *one* agent (with the exception of Section 8.1). The case of multiple agents will be an interesting (but potentially much more difficult) extension of the present work.

These two perspectives on reasoning, as exhibiting nondeterministic behavior and as a dynamic process, are part of a more general framework of levels of abstraction of looking at, and specification of, reasoning, which we will describe below.

1.3 Levels of abstraction

Reasoning processes can be described at many levels of abstraction. The consequence operations mentioned before provide a very high-level description: given some initial knowledge of the agent, they give the final set of conclusions reached, abstracting from how they were formed. On the other hand, we could give a detailed specification of a reasoning system (by giving an implementation in a programming language, or by describing the physical layout of a brain performing the reasoning). But there are intermediate levels of abstraction. Given the fact that often multiple possible sets of conclusions (or *belief sets*) are possible, we could describe the behavior of the agent by giving, for an initial set of beliefs, the possible belief sets (containing possible conclusions), where we still abstract from how these sets were reached. Entering a more detailed description level, we could also indicate the sequence of intermediate states the agent went through in deriving or computing these possible sets of conclusions (we will call such a sequence a *reasoning trace*). But we could also describe a reasoning system that generates these sequences (in the description of systems we could again identify many levels of abstraction, from a description in a high-level specification language, an implementation in a lower-level programming language, to a complete physical description of an agent; we will not consider these abstraction levels here and incorporate all of these abstractions in one level).

These possibilities lead to a hierarchy of levels of abstraction in the description of reasoning (going from more abstract to less abstract):

1. *Specification of a final intended conclusion set*
   Given a set of initial facts, a unique resulting final set of conclusions is specified, disregarding the specific underlying possible (incomplete) belief sets, the specific reasoning patterns leading to them, and the reasoning system generating them.

2. *Specification of a set of intended belief sets*
   Given the initial facts, a set of possible belief sets is specified, abstracting from the specific reasoning patterns leading to them, and the reasoning system generating them.

3. *Specification of a set of intended reasoning traces*
   Given the initial facts, a set of reasoning traces, leading to intended belief sets, is specified, abstracting from the specific reasoning system generating these patterns.

4. *Specification of a reasoning system*
   A reasoning system is specified that, given a set of initial facts as input, can generate the intended reasoning patterns.

Reasoning can be formalized both syntactically and semantically. The discussion above refers to syntactical descriptions (for example with sets of conclusions), but the same notions can be described in semantical terms. We will give a number of examples of descriptions of reasoning at these different levels, both in syntactical and in semantical terms.

Level 1

At level 1, we could give a semantic formalization in terms of a set of intended models for a set of initial facts. This is what happens in preferential logic (see e.g., [Sho87]): given the premises, all models of the premises which are minimal in some preference ordering are selected as intended models. The corresponding set of conclusions is formed by all formulae that are true in all these intended models.

Level 2

As an example of a description at level 2, we could take autoepistemic logic or default logic, in which, given a theory, there can be multiple expansions or extensions. In this case also, there are semantical counterparts, in which a preference ordering on sets of models is given. The minimal sets in the ordering describe the intended conclusions semantically: given such a minimal set of models, the formulae that hold in all models in the set correspond to an extension (see e.g., [BS94], [Voo93]).

Level 3

Specification at level 3 is given in, for instance, argumentation systems (see e.g., [Pol87], [Lou87], [PS96], [Pra97]), where, during the argumentation, arguments in favor and against a fact are given, and counterarguments against arguments. The construction of arguments and counterarguments can be seen as a process. But also the alternative characterization of extensions for default logic in terms of sequences of theories (already given in [Rei80b]) can be seen as a specification of reasoning traces (we will develop this viewpoint in more detail later on; the presentation will be semantical). In dynamic semantics ([Gro95], [Vel96]) the effect of incoming information on the information state of an agent is studied; it is explicitly specified in which order information is coming in. This means the agent goes through a sequence of information states (a new information state arises with every new piece of information coming in): traces are specified in a semantical way. Also in step-logic ([Elg88]), there is an explicit notion of steps in a reasoning (or derivation) process taking place in time. Finally, the cut operator in Prolog is a good example of a dynamic operator influencing the reasoning process. In [Lin97] a semantics of the cut operator in situation calculus is given, where reasoning as an explicit process in time is modeled. We will give a more thorough survey of some of these approaches and a comparison with our own work at a later stage.

Level 4

Specification at level 4 involves describing a system which reasons. This can be a description of an agent in a high-level specification language, a description of a computer program in any programming language, a blueprint of a machine, or even a description of a human being. Specification of a reasoning system occurs frequently within the field of knowledge-based systems and multi-agent systems, see for instance

[BLRT94] for a formally specified agent reasoning about design. In [TT92] a formal architecture of a compositional reasoning system is given, which can perform default reasoning.

Connections

Of course there exist connections between the levels in the sense that from a specification of a lower level of abstraction a specification of each of the higher levels can be determined (in a canonical manner; these connections will be made precise in Chapter 2). For example, given a specification of intended belief sets (level 2), there are ways to arrive at a final set of conclusions (level 1), for example by taking the intersection, or by choosing one of the belief sets. Or, given a set of intended reasoning traces (level 3), one can take a kind of limit, provided that the traces converge in some sense, to arrive at possible belief sets (level 2). The specification at a lower level gives in some sense a refinement or specialization of the specification at the higher level (as in the case of conventional software specifications at different levels of abstraction). Given specifications of two different levels, relative verification should be possible: to establish whether the lower level specification indeed refines the higher level one. At a lower level different specifications can refine the same higher level specification. As a parallel one may think of development of programs using the method of (top down) stepwise refinement, e.g., according to Dijkstra's approach [Dij76]. Note however that other methods (other than top down stepwise refinement) are possible as well.

Much work has been done on formalization and specification of reasoning at levels 1 and 4. By formalization of reasoning we mean that mathematical objects are defined that are an abstraction (at a certain level) of the reasoning. Consequence operations, for example, are mathematical objects that are an abstraction (at level 1) of reasoning. In Chapter 2, we will define mathematical objects that are abstractions of reasoning at lower levels. To describe these objects, we could of course use general mathematical terms (and we often will). However, we will also define specialized (logical) languages for specifying these objects. Such languages will be called *specification languages*. Just as a software specification language allows the user to precisely describe a software system (at a certain level of abstraction), our specification languages allow the user to precisely describe a mathematical object. In this sense, they are no different from any logic with semantics: a statement in a logic 'specifies' its semantics. In this way, propositional logic can be seen as a specification language for sets of valuations (as an example, the formula $p \vee q$ specifies the set of valuations in which either p or q is true). Our specification languages describe the mathematical objects that formalize reasoning at a certain level of abstraction. In this thesis, we will provide approaches to bridge the gap between the highly abstract level 1 and the highly detailed level 4 by developing levels 2 and 3, with most emphasis on level 3.

1.4 Conclusions and related work

In this chapter, we have introduced a hierarchy of abstraction levels at which reasoning (in its broadest sense) can be described. The highest level of abstraction, level 1, was inspired by the work on abstract (nonmonotonic) consequence relations (such as the studies of Gabbay, [Gab85], Shoham, [Sho87], and Kraus, Lehmann, and Magidor [KLM90]) and inference operations (see for example [Mak89] and [Mak94]). The latter paper (but also [Voo93]) already suggests to look at intended belief sets abstractly. Specifying reasoning by giving a reasoning system of course occurs often in Artificial Intelligence.

1.5 Outline of the thesis

We will now give the reader an overview of the further contents of this thesis. In Chapter 2 the first three levels are formalized semantically. For level 3, this formalization involves an operator that assigns sets of reasoning traces to each set of input facts. A language for specifying reasoning on level 3 must therefore be able to describe these traces. In Chapter 3 three example specification languages are treated. One of these, based on temporal logic, will be the subject of the larger part of this thesis. The use of temporal logic is based on the observation that the process of reasoning can be seen as a process taking place *in time*, and that therefore reasoning traces can be seen as a kind of temporal models, in which a state at a certain time point must reflect the information the agent has at that point. Temporal logics of information can be used to specify temporal models which are interpreted as reasoning traces. A number of variants of temporal logic are treated in Chapter 4. In Chapter 5 the usefulness of these temporal logics is established by showing that a number of existing forms of reasoning can be specified in these logics. Restricted versions of these logics are executable, which means that a specification in such a restricted variant can be directly executed (or equivalently, a model can be constructed). This is the content of Chapter 6. The expressiveness (in a more formal logical sense) of two specification languages, one based on temporal logic and one on default logic, is treated in Chapter 7. Then Chapter 8 discusses two applications of the theory developed so far, one of which is the specification and verification of compositional multi-agent systems, and one of which is the formalization of an approximate classification task needed for ecological monitoring. Some logical themes, among which axiomatizability, decidability and complexity of one of the temporal logics, are treated in Chapter 9. Chapter 10 further explores the syntactical side of the formalization of level 2 given in Chapter 2. Finally, in Chapter 11 we summarize the main ideas of this thesis and draw some general conclusions. Also, a perspective on further research is sketched.

Acknowledgments

Much of the research this thesis reports on has been done in cooperation with others, and has already been published. At the end of each chapter, it is indicated whether the material in that chapter has been published before, and where (and this will also indicate with whom – if any – cooperation has taken place).

The idea of distinguishing these 4 levels of abstraction in the description and specification of reasoning appeared in [EHT95] (where level 4 was split in two levels).

Chapter 2

Semantics of Reasoning

In Chapter 1, a framework of four levels of abstraction for the description of reasoning was briefly introduced. A semantical formalization of the first three levels is given in this chapter, along with a number of examples.

We will start the formal description of the three levels of abstraction by introducing the basic ingredients needed in all three. First of all, there must be a language in which information about the reasoning domain of the agent can be expressed. So, we will assume that the reasoning agent has a language $\mathcal{L}$, in which the information it has (beliefs or knowledge) can be stated. Sets of initial facts are also given in this language. Although no special requirements have to be imposed on this language, it could for instance be a language of propositional logic, or predicate logic, or modal logic. If $\mathcal{L}$ is a logical language, there is often a notion of (classical) consequence or provability. In such cases, an operator which assigns to a set of formulae the set of all of its consequences, will be denoted by $C_{\mathcal{L}}$.

As mentioned before, the agent's mental state must incorporate the information the agent has (knows, or believes). The part of the mental state holding this information, will be called an *information state*. Therefore, the second assumption is that there exists a set, IS, of possible information states of the agent. Since these states hold the information of the agent, there must be a notion capturing the fact that a sentence in the language is contained in an information state. We will assume the existence of an operator Th, which assigns to each information state the set of sentences it holds: $\mathrm{Th} : \mathsf{IS} \rightarrow \mathcal{P}(\mathcal{L})$, where $\mathcal{P}(\mathcal{L})$ denotes the powerset of $\mathcal{L}$. For an information state M, the set $\mathrm{Th}(M)$ is called the *theory of* M.

The beliefs of an agent change as a consequence of internal mental operations the agent performs. These operations include reasoning, but also the agent may communicate, perform observations and revisions, and it may sometimes simply forget things. In one situation an agent may have more (or less) information than in another. To be able to compare information in different situations, we will assume that the set of information states, IS, is equipped with a partial order (that is, a relation satisfying reflexivity, antisymmetry and transitivity), denoted $\preceq$. For two information states M and N, if $M \preceq N$, this will mean that in state N, the agent has more information than (or the same information as) in state M. Often, the operator Th will be *monotone*, meaning that $\mathrm{Th}(M) \subseteq \mathrm{Th}(N)$ whenever $M \preceq N$. This will not be required, however. (It may be the case that the ordering $\preceq$ is based on information about the world only, whereas $\mathrm{Th}(N)$ also contains *epistemic* information: information about which information is known; this kind of information is not monotone.)

A second requirement is that it is possible to aggregate the information from an increasing sequence of information states. If the agent traverses a linearly ordered sequence of states, the information it has is increasing. Therefore, we can consider *all* the information it gains during the reasoning. There should be an information state that holds precisely all this information. Formally, we will require that each linearly ordered (with respect to $\preceq$) subset A of information states has a least upper bound, $\mathrm{lub}(A)$ in IS. Thirdly, given a number of information states, the agent may sometimes wish to keep only the information common to all of them. Hence, it will be assumed that any set of information states A, has a greatest lower bound, $\mathrm{glb}(A)$. For ease of reference, the requirements are listed below.

Definition 2.1 (Information state frame) An *information state frame* is a tuple $\langle \mathcal{L}, C_{\mathcal{L}}, \mathsf{IS}, \preceq, \mathrm{Th} \rangle$, where

- $\mathcal{L}$ is a set, called a *language*.
- $C_{\mathcal{L}}$ is a function $\mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathcal{L})$, satisfying
    1. $X \subseteq C_{\mathcal{L}}(X)$ (*inclusion*)
    2. $C_{\mathcal{L}}(C_{\mathcal{L}}(X)) = C_{\mathcal{L}}(X)$ (*idempotence*)
    3. $X \subseteq Y \Rightarrow C_{\mathcal{L}}(X) \subseteq C_{\mathcal{L}}(Y)$ (*monotony*)
- IS is a set, the elements of which are called *information states*.
- $\preceq$ is a partial order on IS, such that for each non-empty $A \subseteq \mathsf{IS}$, the greatest lower bound $\mathrm{glb}(A)$ of A with respect to $(\mathsf{IS}, \preceq)$ exists, and if $(A, \preceq)$ is a linear order, there is a least upper bound $\mathrm{lub}(A)$ with respect to $(\mathsf{IS}, \preceq)$. The ordering $\preceq$ is called the *information order*.
- Th is a function $\mathsf{IS} \to \mathcal{P}(\mathcal{L})$. For $M \in \mathsf{IS}$, the set $\mathrm{Th}(M)$ is called the *theory of* M.

We will give some examples of information state frames. In all of them, let P be a finite or denumerably infinite set of of propositional atoms.

Definition 2.2 (Two-valued states) The language $\mathcal{L}$ is equal to P, and $C_{\mathcal{L}}$ is the identity function. A (two-valued) propositional valuation is a function $m : P \to \{0, 1\}$.

1. The set IS^{2val} consists of all valuations.

2. The truth-values 0 (false) and 1 (true) are ordered as follows: $0 \leq 1$ and $0 \leq 0$, $1 \leq 1$. For two valuations m and n we define $m \preceq n$ if and only if $m(p) \leq n(p)$ for all $p \in P$.

3. Define $\mathrm{Th} : \mathsf{IS}^{2val} \to \mathcal{P}(\mathcal{L})$ by $\mathrm{Th}(m) = \{p \in P \mid m(p) = 1\}$.

It is straightforward to check that $\preceq$ is a partial order and that greatest lower bounds and least upper bounds (for any set, not just for linearly ordered sets) exist: just take the minimum (or maximum) per atom. The frame of two-valued states is often used under the closed world assumption: the facts that have been verified are true, all others are false by default.

This asymmetry between positive and negative information does not occur in three-valued states.

Definition 2.3 (Three-valued states) Let $\mathcal{L}$ be the propositional language based on P.

1. A *partial* (or *three-valued*) model for the language $\mathcal{L}$ is a function $m : P \to \{0, 1, u\}$. This assignment can be extended to arbitrary propositional formulae according to the following tables (these are the Strong Kleene semantics):

$\neg$	
0	1
1	0
u	u

$\wedge$	0	1	u
0	0	0	0
1	0	1	u
u	0	u	u

$\vee$	0	1	u
0	0	1	u
1	1	1	1
u	u	1	u

$\to$	0	1	u
0	1	1	1
1	0	1	u
u	u	1	u

IS^{3val} denotes the set of all partial models for $\mathcal{L}$.

2. The operator $C_{\mathcal{L}}$ is strong semantic consequence: $C_{\mathcal{L}}(X) = \{\varphi \mid$ for all partial models m, if $m(\psi) = 1$ for all $\psi \in X$, then $m(\varphi) = 1\}$.

3. The truth-values 0 (false), 1 (true), and u (unknown) are ordered as follows: $u \leq 0$, $u \leq 1$, and $u \leq u$, $0 \leq 0$, $1 \leq 1$. For two partial models m and n, we define: $m \preceq n \Leftrightarrow m(p) \leq n(p)$ for all $p \in P$.

4. Define $\mathrm{Th} : \mathsf{IS}^{3val} \to \mathcal{P}(\mathcal{L})$ by $\mathrm{Th}(m) = \{\varphi \in \mathcal{L} \mid m(\varphi) = 1\}$.

It is straightforward to check that $\preceq$ is a partial order. For a non-empty set of partial models A, the greatest lower bound, $\mathrm{glb}(A)$, is given by:

$$\mathrm{glb}(A)(p) = \begin{cases} 1, & \text{if } m(p) = 1 \text{ for all } m \in A; \\ 0, & \text{if } m(p) = 0 \text{ for all } m \in A; \\ u & \text{otherwise.} \end{cases}$$

For a linearly ordered set $A \subseteq \mathsf{IS}$, the least upper bound, $\mathrm{lub}(A)$, is given by:

$$\mathrm{lub}(A)(p) = \begin{cases} 1, & \text{if } m(p) = 1 \text{ for some } m \in A; \\ 0, & \text{if } m(p) = 0 \text{ for some } m \in A; \\ u & \text{otherwise.} \end{cases}$$

Since in a linearly ordered set, no two different partial models m, n can exist with $m(p) = 1$ and $n(p) = 0$ for some $p \in P$, the lub exists. If $m \preceq n$, then it holds $\mathrm{Th}(m) \subseteq \mathrm{Th}(n)$; this is a consequence of the persistence theorem for partial logic.

Proposition 2.4 (Persistence) Suppose $n \preceq m$ are partial models. For every propositional formula φ, the following holds:

$$\begin{array}{lcl} n(\varphi) = 1 & \Rightarrow & m(\varphi) = 1 \\ n(\varphi) = 0 & \Rightarrow & m(\varphi) = 0. \end{array}$$

One disadvantage of partial models is their inability to express disjunctive information without committing to either of the disjuncts. That is, if (under the strong Kleene semantics) $m(\varphi \vee \psi) = 1$, then either $m(\varphi) = 1$ or $m(\psi) = 1$ (or both). So an agent cannot believe a disjunction without believing one of the disjuncts. Furthermore, tautologies are not necessarily true in a partial model, for example, if $m(p) = u$, then $m(p \vee \neg p) = u$. In some situations, this may be a disadvantage of partial logic, but sometimes this may be desired (not all agents have to know all tautologies). An approach without these (arguable) disadvantages, uses sets of two-valued models.

Definition 2.5 (Epistemic states)

1. The language $\mathcal{L}$ is the propositional language based on P, with the standard consequence operator Cn of propositional logic.
2. An information state is a set of propositional valuations. The set of these information states is denoted as IS^{ep}.
3. The ordering $\preceq$ on the set of information states IS^{ep}, is defined by: $M \preceq N \Leftrightarrow N \subseteq M$.
4. The function Th is defined by: $\mathrm{Th}(M) = \{\varphi \in \mathcal{L} \mid m \models \varphi \text{ for all } m \in M\}$.

Since set-inclusion is a partial order, its converse ($\preceq$) is also a partial order. The least upper bound of a (not necessarily linearly ordered) set A of information states, is given by $\mathrm{lub}(A) = \bigcap_{M \in A} M$. The greatest lower bound of a set A of information states, is given by $\mathrm{glb}(A) = \bigcup_{M \in A} M$. As was the case for three-valued models, the operator Th is monotone: we have $\mathrm{Th}(M) \subseteq \mathrm{Th}(N)$ whenever $M \preceq N$.

Notice that for any information state M, its theory contains all tautologies. Furthermore, if we define $M = \{m \mid m \models p \text{ or } m \models q\}$, then $p \vee q \in \mathrm{Th}(M)$, whereas neither $p \in \mathrm{Th}(M)$ nor $q \in \mathrm{Th}(M)$.

The previous examples (Definitions 2.2, 2.3 and 2.5) involve semantical notions (models of a language), but this is not a requirement, as the next example illustrates.

Definition 2.6 (Syntactic states) Let $\mathcal{L}$ again be a propositional language with consequence operator Cn. Let IS^{syn} consist of all sets (of propositional formulae of $\mathcal{L}$) closed under propositional consequence, that is, $\mathsf{IS}^{syn} = \{S \subseteq \mathcal{L} \mid Cn(S) = S\}$, where $Cn(S) = \{\varphi \in \mathcal{L} \mid S \models \varphi\}$. Define the ordering $\preceq$ by $S \preceq T \Leftrightarrow S \subseteq T$. The function Th is the identity: $\mathrm{Th}(S) = S$.

Given these definitions, for a (not necessarily linearly ordered) set $A \subseteq \mathsf{IS}^{syn}$ it holds that $\mathrm{lub}(A) = Cn(\bigcup_{S \in A} S)$ and $\mathrm{glb}(A) = \bigcap_{S \in A} S$.

In a sense, the syntactic states of Definition 2.6 are the syntactical counterpart to the epistemic states of Definition 2.5. Let us denote the function Th of IS^{ep} by Th_{ep} and the one of IS^{syn} by Th_{syn}. Given an epistemic state $M \in \mathsf{IS}^{ep}$, define $S(M) = \mathrm{Th}_{ep}(M)$. It is easy to see that $Cn(S(M)) = S(M)$ so that $S(M) \in \mathsf{IS}^{syn}$. Furthermore, $\mathrm{Th}_{syn}(S(M)) = \mathrm{Th}_{ep}(M)$. Going the other way, let $S \in \mathsf{IS}^{syn}$ be a syntactic state. Define $M(S) = Mod(S)$, where $Mod(S) = \{m \mid m \text{ is a valuation such that } m \models \varphi \text{ for all } \varphi \in S\}$. Then $M(S) \in \mathsf{IS}^{ep}$ and $\mathrm{Th}_{ep}(M(S)) = \mathrm{Th}_{syn}(S)$. This means that any description of reasoning involving epistemic states, can be phrased equivalently using syntactic states, and vice versa.

We are now ready to formalize the first three levels of abstraction.

2.1 Level 1: final conclusion set

A description of reasoning at level 1 involves specification of a final set of conclusions, given a set of initial facts. This can be formalized by an operator that assigns information states (holding these final conclusions) to sets of formulae.

Definition 2.7 (Final belief state operator) A *final belief state operator (FBSO)* is a function $\Phi : \mathcal{P}(\mathcal{L}) \to \mathsf{IS}$.

Given a set of initial facts $X \subseteq \mathcal{L}$, the reasoning process of the agent gives rise to a unique information state, $\Phi(X)$, containing the conclusions $\mathrm{Th}(\Phi(X))$. Many examples of FBSOs occur in the literature, of which three are described below.

One of the most well-known FBSOs is the consequence operator Cn of a classical logic. Given the information state frame of Definition 2.6, the consequence operator of propositional logic is easily seen to be a final belief state operator.

A preferential logic ([Sho87], [Sho88], [Mak94]) consists of a classical logic, given by a language $\mathcal{L}$, a model class Mod, and a satisfaction relation $\models \subseteq \mathcal{L} \times \mathsf{Mod}$, together with a partial order $\sqsubset$ on Mod. A model $m \in \mathsf{Mod}$ is called a *minimal model* of a set of formulae $A \subseteq \mathcal{L}$, denoted $m \models_{\sqsubset} A$, if $m \models A$ (meaning that $m \models \varphi$ for all $\varphi \in A$), and there is no other model in Mod which is smaller than m in the ordering $\sqsubset$ which satisfies A.

Example 2.8 (Preferential logic) Let $\langle \mathcal{L}, Cn, \mathsf{IS}^{ep}, \preceq, \mathrm{Th}\rangle$ be the information state frame of Definition 2.5. Given a preferential logic $(\mathcal{L}, \mathsf{Mod}, \models, \sqsubset)$, define $\Phi_{\sqsubset}(X) = \{m \in \mathsf{Mod} \mid m \models_{\sqsubset} X\}$.

Based on this FBSO, a nonmonotonic entailment relation $\mid\!\sim$ can be defined by $A \mid\!\sim \varphi \Leftrightarrow \varphi \in \mathrm{Th}(\Phi_{\sqsubset}(A))$. Although the technical details differ, the notion of preferential logic and the entailment relation defined here, are very similar to those in [KLM90] (see also Sections 9.2 and 9.3). There are also examples of syntactically defined FBSOs (these will be studied more thoroughly in Chapter 10).

The field of belief revision is concerned with the addition and removal of information to or from a set of beliefs ([AGM85] is the most influential paper in this area). This phenomenon is modeled by *expansion, contraction* and *revision operators.* An expansion operator $+$ is a function that assigns to a set of beliefs K, and a formula φ, a new set of beliefs, $K + \varphi$, the addition of φ to K. Removal of information is modeled by a contraction operator $\dot{-}$, that assigns to a set of beliefs K and a formula φ, a new set of beliefs $K \dot{-} \varphi$, which is the result of removing φ from K. Expanding a set of beliefs K by a formula φ may result in an inconsistent belief set (if $\neg\varphi$ was a member of K). Revision operators also add a formula to a set of beliefs, but without causing inconsistency. [AGM85] gives a number of postulates that these operators should satisfy, commonly called the AGM postulates for belief revision (see Example 10.44, where the postulates for contraction are given).

Example 2.9 (Belief revision) Let IS^{syn}, $\preceq$ and Th be as in Definition 2.6, and let $+$ be an expansion operator and $\dot{-}$ be a contraction operator, satisfying the AGM postulates. For a formula $\varphi \in \mathcal{L}$, define the following two operators:

$$\begin{aligned}\Phi_{+\varphi}(X) &= Cn(X) + \varphi\\ \Phi_{-\varphi}(X) &= Cn(X) \dot{-} \varphi.\end{aligned}$$

The AGM postulates ensure that the result of expansion or contraction of an information state, is again an information state (specifically, it is closed under propositional consequence).

The definition of a final belief state operator is quite general, and admits all sorts of operators, however ill-behaved. A number of requirements can be posed on FBSOs that ensure better behavior.

Definition 2.10 (Properties of FBSOs) Let $\Phi : \mathcal{P}(\mathcal{L}) \to \mathsf{IS}$ be a final belief state operator.

1. The operator Φ satisfies *inclusion* if for all $X \subseteq \mathcal{L}$: $X \subseteq \mathrm{Th}(\Phi(X))$;

2. The operator Φ satisfies *monotony* if for all $X, Y \subseteq \mathcal{L}$: $X \subseteq Y \Rightarrow \Phi(X) \preceq \Phi(Y)$.

3. The operator Φ satisfies *cautious monotony* if for all $X, Y \subseteq \mathcal{L}$: $X \subseteq Y \subseteq \mathrm{Th}(\Phi(X)) \Rightarrow \Phi(X) \preceq \Phi(Y)$.

4. The operator Φ satisfies *cut* if for all $X, Y \subseteq \mathcal{L}$: $X \subseteq Y \subseteq \mathrm{Th}(\Phi(X)) \Rightarrow \Phi(Y) \preceq \Phi(X)$.

5. The operator Φ satisfies *invariance* if for all $X \subseteq \mathcal{L}$: $\Phi(X) = \Phi(C_{\mathcal{L}}(X))$.

The properties given above are certainly not meant as mandatory requirements for any FBSO. The operator $\Phi_{-\varphi}$ of Example 2.9, for example, should remove part of the input, and should therefore not satisfy inclusion. The operator associated with a preferential logic (which is meant to describe nonmonotonic reasoning) should in general not satisfy monotony. Let us review the examples to see which properties are satisfied by them.

It is well-known that the operator Cn of propositional logic satisfies all properties. The operator $\Phi_{\sqsubset}$ of a preferential logic satisfies inclusion, invariance and cut. Under certain restrictions (*smoothness*; see [KLM90]) it satisfies cautious monotony. Monotony is in general not satisfied, although it is possible, depending on the partial order $\sqsubset$. The expansion operator $\Phi_{+\varphi}$ satisfies all properties (it is shown in [AGM85] that the only expansion operator possible assigns $Cn(X \cup \{\varphi\})$ to X and φ). The contraction operator $\Phi_{-\varphi}$ does not satisfy inclusion, but satisfies invariance, cautious monotony, and cut (the properties of monotony, cautious monotony, and cut are usually defined only for operators satisfying inclusion). It does not need to satisfy monotony.

An FBSO Φ based on the information states of Definition 2.5, is called a *model operator* in [EHT95] (see also [Her94]) if it satisfies inclusion.

2.2 Level 2: set of belief sets

A description of reasoning on level 2 involves specification of possible belief sets, given the initial facts. This can be formalized by an operator that assigns a set of information states (each holding the information of one possible belief set) to each set of formulae.

Definition 2.11 (Multiple belief state operator) A *multiple belief state operator (MBSO)* is a function $\Gamma : \mathcal{P}(\mathcal{L}) \rightarrow \mathcal{P}(\mathsf{IS})$.

Given a set of initial facts $X \subseteq \mathcal{L}$, the reasoning process of the agent may lead to a number of possible information states, which together form $\Gamma(X)$. These information states are possible views on the domain, sanctioned by the reasoning, given the initial facts. Such operators have not been studied extensively in the literature (but see [Mak94], where extension family operators are defined and [Voo93] where extension operations are defined), even though the phenomenon that a set of initial facts may have more than one intended outcome, occurs often in nonmonotonic reasoning. We will give some examples.

Example 2.12 (Default logic) A thorough introduction to default logic will be given in Section 3.1, and the reader unfamiliar with default logic is invited to read that section first. Let D be a set of defaults. For $X \subseteq \mathcal{L}$, let $\mathsf{Ext}(D, X)$ denote the set of Reiter extensions of the default theory $\langle D, X\rangle$. Define the operator $\mathcal{B}^{dl}_D$ by $\mathcal{B}^{dl}_D(X) = \mathsf{Ext}(D, X)$. Given the information state frame of syntactic states of Definition 2.6, this defines a multiple belief state operator. A semantic version Γ^{dl}_D of this operator can be defined by $\Gamma^{dl}_D(X) = \{Mod(E) \mid E \in \mathsf{Ext}(D, X)\}$ (see the discussion after Definition 2.6). This semantic operator Γ^{dl}_D can also be given a direct definition, in the spirit of preferential logic. A preference relation $<_D$ on *sets* of models is introduced in [Eth87], along with a notion of D-stability of a set of models. In [Eth87] it is shown that $\Gamma^{dl}_D(X) = \{M \subseteq Mod(X) \mid M$ is $<_D$-minimal and D- stable$\}$.

In this same fashion, autoepistemic logic (see [Moo85], [Kon94]) gives rise to an MBSO. Again, we defer an introduction of autoepistemic logic to later (Section 5.6).

Example 2.13 (Autoepistemic logic) Consider the information state frame of Definition 2.6, but with a propositional *modal* language $\mathcal{L}$, and Cn denotes modal provability in the minimal modal logic K. Define the operator $\mathcal{B}^{ael}$ by $\mathcal{B}^{ael}(X) = \{S \subseteq \mathcal{L} \mid S$ is an autoepistemic expansion of $X\}$.

The last example is again taken from belief revision.

Example 2.14 (Belief revision) Let the information state frame again be the one of Definition 2.6. Given a set A of propositional formulae that is closed under propositional consequence, and a formula φ, define $A \perp \varphi = \{B \mid B \subseteq A\backslash\{\varphi\}, Cn(B) = B$ and B is minimal with respect to these requirements $\}$. Define the MBSO $\Gamma_{-\varphi}$ by $\Gamma_{-\varphi}(X) = Cn(X) \perp \varphi$. The information states in $\Gamma_{-\varphi}(X)$ all represent ways of removing φ from X, while retaining as much of X as possible. A priori, there is no preference for any of these possibilities (if there is more than one).

One can formulate properties of multiple belief state operators, as was done for final belief state operators previously. Some of these properties are generalizations of properties of FBSOs, but others are specific for MBSOs.

Definition 2.15 (Properties of MBSOs) Let $\Gamma : \mathcal{P}(\mathcal{L}) \rightarrow \mathcal{P}(\mathsf{IS})$ be a multiple belief state operator.

1. The operator Γ satisfies *inclusion* if for all $X \subseteq \mathcal{L}$: $X \subseteq \mathrm{Th}(M)$ for all $M \in \Gamma(X)$;
2. The operator Γ satisfies *non-inclusiveness* if for all $X \subseteq \mathcal{L}$ and $M, N \in \Gamma(X)$: $M \preceq N \Rightarrow M = N$;
3. The operator Γ satisfies *invariance* if for all $X \subseteq \mathcal{L}$: $\Gamma(X) = \Gamma(C_{\mathcal{L}}(X))$.

The properties of monotony, cautious monotony and cut can also be generalized, see Chapter 10. The operators of Example 2.12, 2.13 and 2.14 all satisfy invariance. Inclusion is satisfied by the operators of default logic and autoepistemic logic, not by the belief revision operator. Non-inclusiveness is not satisfied by the operator of autoepistemic logic, but it is satisfied by the other two.

Multiple belief state operators give a description of reasoning at level 2. There are a number of well-established possibilities of abstraction of an MBSO, yielding a level 1 description. The sceptic approach retains only that information common to all possible information states. The choice approach chooses one of the information states. Both possibilities fall under a more general scheme we will describe below.

Definition 2.16 (Selection operator) A *selection operator* is a function $s : \mathcal{P}(\mathsf{IS}) \to \mathcal{P}(\mathsf{IS})$ such that

- $s(A) \subseteq A$,
- $s(A) \neq \emptyset$ whenever $A \neq \emptyset$.

If for every $A \subseteq \mathsf{IS}$, the set $s(A)$ contains exactly one element, we call s *single-valued*.

Definition 2.17 (FBSO associated to an MBSO)

1. For a multiple belief state operator $\Gamma : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathsf{IS})$, the *associated* final belief state operator $\Phi_\Gamma : \mathcal{P}(\mathcal{L}) \to \mathsf{IS}$ is defined by $\Phi_\Gamma(X) = \mathrm{glb}(\Gamma(X))$. Let a selection operator s be given. Then the final belief state operator Φ_Γ^s *associated* to Γ *and* s, is defined by $\Phi_\Gamma^s(X) = \mathrm{glb}(s(\Gamma(X)))$.
2. For a given FBSO $\Phi : \mathcal{P}(\mathcal{L}) \to \mathsf{IS}$, define $\Gamma_\Phi = \{\Gamma \mid \Gamma \text{ is an MBSO with } \Phi_\Gamma = \Phi\}$.

The operator Φ_Γ is in a sense sceptical: it only yields information common in all possibilities of Γ, but as much of this as possible. The conclusions given by the FBSO associated to the operator Γ^{dl} of default logic, are also called the *sceptical* conclusions in default logic. The operator $\Phi_{\Gamma_{-\varphi}}$ associated to the MBSO of Example 2.14 yields a special kind of contraction, called *full meet contraction* in [AGM85].

Before performing this operation of taking the information common in the possible information states, a selection operator allows us to consider only a subset of possibilities. This subset can consist of the 'best' possible information states, that is, most interesting, most descriptive, most probable, most preferred, etc. By taking the identity function as selection operator, we again get Φ_Γ. If we take a *single-valued* selection operator s, i.e., one which assigns a singleton information state to each non-empty set of information states, we get the choice approach. If $s(\Gamma(X)) = \{Y\}$, then $\Phi_\Gamma^s(X) = \mathrm{glb}(s(\Gamma(X))) = \mathrm{glb}(\{Y\}) = Y \in \Gamma(X)$.

In general, for an FBSO Φ, the set Γ_Φ may contain many MBSOs. It contains at least one trivial operator Γ, defined by $\Gamma(X) = \{\Phi(X)\}$.

Given a multiple belief state operator Γ that satisfies certain properties, one can ask whether these are transferred to corresponding properties of Φ_Γ^s. Some of these correspondences are given below (see also Chapter 10).

Proposition 2.18 (Transfer of properties from level 2 to level 1) Let a multiple belief state operator Γ, a selection operator s and their associated final belief state operator Φ_Γ^s be given.

1. If Γ satisfies invariance, then Φ_Γ^s satisfies invariance.
2. Suppose the operator Th is monotone ($N \preceq M \Rightarrow \mathrm{Th}(N) \subseteq \mathrm{Th}(M)$). If Γ satisfies inclusion, then Φ_Γ^s satisfies inclusion.

Proof: Straightforward. □

2.3 Level 3: set of reasoning traces

A description of reasoning processes at level 3 involves specification of a set of reasoning traces. Reasoning is viewed, on level 3, as a stepwise process. The agent starts, having a certain information state, and performs some operation(s) on this information state, yielding a new information state. In this new state, the agent may again perform some operation(s). This results in a sequence of information states. The operation performed on an information state may be an elementary inference step in some classical logic, it may be application of a default rule (or some other nonmonotonic inference), but it may also be an observation or a communication act. During the reasoning, the world the agent reasons about, may either change or remain the same. If the world remains the same, the agent's knowledge about this world changes as a result of obtaining new information, drawing new conclusions and making new default assumptions. If the world changes, then this provides additional reasons for the agent's knowledge to change: observations (and communications) performed before may later be invalidated by new observations.

A reasoning process may go on forever, or it may terminate. To make the formalization uniform, we will nevertheless represent such a finite process as an infinite sequence, but one in which from a certain index onwards, nothing changes. The natural numbers (0, 1, 2, ...) are denoted by $\mathbb{N}$.

Definition 2.19 (Reasoning trace)

1. A *reasoning trace* $\mathcal{M}$ is a sequence $(\mathcal{M}_i)_{i\in\mathbb{N}}$ where $\mathcal{M}_i \in \mathsf{IS}$ for all $i \in \mathbb{N}$. Viewing $\mathcal{M}$ as a function from $\mathbb{N}$ to IS, we will sometimes denote $\mathcal{M}_i$ as $\mathcal{M}(i)$. The set of all reasoning traces is denoted as $\mathsf{Traces}(\mathsf{IS})$.
2. A reasoning trace $\mathcal{M}$ is called *end-conservative* if there exists an index $k \in \mathbb{N}$ such that $\mathcal{M}_i \preceq \mathcal{M}_{i+1}$ for all $i \geq k$ (where $\preceq$ is the ordering of IS). If $k = 0$, then $\mathcal{M}$ is simply called *conservative.*
3. A reasoning trace $\mathcal{M}$ is called *eager* if for all $s \in \mathbb{N}$: $\mathcal{M}_s = \mathcal{M}_{s+1} \Rightarrow \mathcal{M}_s = \mathcal{M}_t$ for all $t > s$.

In an end-conservative reasoning trace, no information is lost from a certain index onwards. This means that the agent may not forget or revise its knowledge. Finite processes (in which $\mathcal{M}_i$ is constant from some index onwards) are certainly end-conservative. Eager processes model reasoning of an agent that never waits: if at a certain transition the information state is not changed (the agent has not performed any reasoning) then this must mean that the agent is finished.

For many reasoning processes, we are interested in the conclusions reached during this process. This is not always the case: for an agent continually performing observations (or communications) in a changing world, conclusions reached during some stage may be incorrect later (when the world has changed). For a finite process, however, there is a clear notion of the final conclusions. The exact property a reasoning process has to satisfy for such a notion of "final conclusions", or "limit model" to make sense, is end-conservativity.

Definition 2.20 (Limit model) Let $\mathcal{M}$ be an end-conservative trace, where k is an index as mentioned in Definition 2.19. The *limit model* $\lim\, \mathcal{M}$ of $\mathcal{M}$, is defined as: $\lim\, \mathcal{M} = \mathrm{lub}(\{\mathcal{M}_i \mid i \geq k\})$.

Note that for end-conservative traces, $\lim\, \mathcal{M}$ is well-defined as $\{\mathcal{M}_i \mid i \geq k\}$ is a linearly ordered set with respect to the ordering on IS. The least upper bound of such a set is assumed to exist by Definition 2.1. The definition is independent of the particular index k taken.

Given a set of initial facts, the reasoning trace certainly does not have to be uniquely determined. In the presence of default knowledge, there are often multiple choices of which default conclusions to add. But also observations or communication may lead to different traces, based on what is observed (communicated). The (internal) reasoning behavior of an agent can be captured by the set of all possible reasoning traces it can generate.

Definition 2.21 (Reasoning frame operator) A *reasoning frame operator* is a function $\mathcal{T} : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathsf{Traces}(\mathsf{IS}))$, such that for all $X \subseteq \mathcal{L}$, for all traces $\mathcal{M} \in \mathcal{T}(X)$: $\mathrm{Th}(\mathcal{M}_0) = C_{\mathcal{L}}(X)$. If $\mathcal{T}(X)$ consists of (end-)conservative traces for each

$X \subseteq \mathcal{L}$, then $\mathcal{T}$ is called *(end-)conservative.* If $\mathcal{T}(X)$ consists of eager traces for each $X \subseteq \mathcal{L}$, then $\mathcal{T}$ is called *eager.*

The requirement on the function $\mathcal{T}$ makes sure that a trace starts with the information state holding the input facts.

Reasoning frames operators are generated by any (step-wise) form of reasoning. We will give some examples.

Example 2.22 (Proof systems) Consider a classical Hilbert-style proof system, with axioms and derivation rules. A proof in such a system consists of a sequence of formulae, where each formula is either an axiom, or the result of the application of a derivation rule to some formula(e) earlier in the sequence. Let the information states be sets of formulae (not necessarily closed under classical consequence), with the ordering of set inclusion. The closure operation is the identity function. In the reasoning trace corresponding to the construction of such a proof, the information state (of the agent) at a certain index consists of the formulae in the proof until and including the formula derived at that index. The reasoning trace is conservative (as the sets of derived formulae form a non-decreasing sequence); as it is not forbidden to prove a formula twice in the same proof, the trace need not be eager (although one could restrict oneself to eager traces). Viewing this as a reasoning process without input, one could define a reasoning frame operator that assigns the set of all traces as defined above, to the empty set, and assigns the empty set of traces to all other input sets. Alternatively, one could see the axioms as the input, or one could allow extra premises to be input. The traces would then start with an information state in which all axioms (all extra premises) are known to the agent. The traces are further constructed as sketched above. See Section 5.4 for a semantical treatment of proof systems.

Another example of a form of reasoning is design.

Example 2.23 (Design) Consider a software agent that gets as its input a partial design (of, for instance, an artifact; this partial design may be empty), together with a number of requirements the complete design has to fulfill. The task of the agent is to complete the partial design, fulfilling the requirements. At each reasoning step, the agent may instantiate a parameter of the design (such as the height of the object), and check if no requirement has been violated yet. Also, the agent may decide that the requirements are too strict (or contradictory), and communicate with its user to negotiate about the requirements. A run of this system, starting with the input partial design together with the requirements, and yielding a complete design, can be formalized as a reasoning trace. Usually, there is more than one possible complete design, so we can take the set of all reasoning traces corresponding to a run of the system leading to a complete design. A reasoning frame operator is defined by assigning to a set of inputs (a partial design and requirements), the set of all traces corresponding to a run of the system on this input (see [BLRT94] and [BLT96]).

In Chapter 3, some more examples are treated.

Besides the properties of individual traces mentioned in Definition 2.19, there are other interesting properties of reasoning frame operators.

Definition 2.24 (Properties of reasoning frame operators) Let $\mathcal{T}$ be an end-conservative reasoning frame operator.

1. The operator $\mathcal{T}$ satisfies *non-inclusiveness*, if for all $X \subseteq \mathcal{L}$ and $\mathcal{M}, \mathcal{N} \in \mathcal{T}(X)$:
$$\lim \mathcal{M} \preceq \lim \mathcal{N} \Rightarrow \lim \mathcal{M} = \lim \mathcal{N}.$$
2. The operator $\mathcal{T}$ satisfies *uniqueness of traces*, if for all $X \subseteq \mathcal{L}$ and $\mathcal{M}, \mathcal{N} \in \mathcal{T}(X)$:
$$\lim \mathcal{M} = \lim \mathcal{N} \Rightarrow \mathcal{M} = \mathcal{N}.$$
3. The operator $\mathcal{T}$ satisfies *invariance* if for all $X \subseteq \mathcal{L}$: $\mathcal{T}(X) = \mathcal{T}(C_{\mathcal{L}}(X))$.

The first property is analogous to the property with the same name for MBSOs, and the second property specifies that from an initial state, there can not be two different ways of reaching the same conclusions. Again, these properties are certainly not supposed to hold for all reasoning frame operators.

Reasoning frame operators describe reasoning at abstraction level 3. For traces which are not end-conservative, there is no (natural) notion of a limit information state, so it makes no sense to aim at a level 2 description of the same reasoning process. For frames consisting of end-conservative traces, there is a natural way of abstracting a level 3 description into a description on level 2.

Definition 2.25 (MBSO associated to reasoning frame operator)

1. Let $\mathcal{T}$ be an end-conservative reasoning frame operator. Define the MBSO $\Gamma_{\mathcal{T}}$ by
$$\Gamma_{\mathcal{T}}(X) = \{\lim \mathcal{M} \mid \mathcal{M} \in \mathcal{T}(X)\}.$$
2. Given an MBSO Γ, define $\mathcal{T}_\Gamma = \{\mathcal{T} \mid \mathcal{T}$ is an end-conservative reasoning frame operator such that $\Gamma_{\mathcal{T}} = \Gamma\}$.

Given an end-conservative reasoning frame operator, there is a unique associated MBSO. Going the other way, there are in general many different sequences leading from an initial state to a final state, so $\mathcal{T}_\Gamma$ may contain many reasoning frames. There is (as was the case when going from FBSOs to MBSOs) a trivial way of associating a trace with a pair (X, M) where $M \in \Gamma(X)$: the first state in the trace is an information state N capturing the initial facts ($\mathrm{Th}(N) = C_{\mathcal{L}}(X)$), and all conclusions of M are drawn in one step: all other states in the trace are equal to M. Assigning to each $X \subseteq \mathcal{L}$ the set of such traces for each pair (X, M) with $M \in \Gamma(X)$, yields an end-conservative reasoning frame operator in $\mathcal{T}_\Gamma$.

There are again correspondences between properties of a reasoning frame operator, and properties of the associated MBSO.

Proposition 2.26 (Transfer of properties from level 3 to level 2) Let $\mathcal{T}$ be an end-conservative reasoning frame operator.

1. If $\mathcal{T}$ satisfies non-inclusiveness, then $\Gamma_{\mathcal{T}}$ satisfies non-inclusiveness;
2. If $\mathcal{T}$ satisfies invariance, then $\Gamma_{\mathcal{T}}$ satisfies invariance.
3. If $\mathcal{T}$ is conservative and Th is monotone then $\Gamma_{\mathcal{T}}$ satisfies inclusion.

Both the space $\mathcal{T}_\Gamma$ defined above and the space Γ_Φ (of MBSOs whose associated MBSO is Φ) may be quite large. One can define a parameterization of these spaces to gain more insight in their structure (see [EHT95]).

2.4 Conclusions and related work

In this chapter, we have given a semantical formalization of the first three levels of abstraction of Chapter 1. That is, we have given a class of (mathematical) objects that are abstractions (at different levels) of the processes of reasoning that may occur in the real world. This involved the notion of an information state, but as the exact nature of these states is not essential, we have left them abstract. Examples were given of information states, and of FBSOs, MBSOs and reasoning frame operators. When we take the information states to be sets of formulae, then FBSOs are just consequence operations (see [Mak94]). The *extension family functions* of [Mak94] and the *extension operations* of [Voo93] are MBSOs under the same information state frame. Our selection functions are briefly mentioned as *choice functions* in [Mak94], although the choice mechanism in particular nonmonotonic logics occurred earlier (as, for instance, for default logic, [Rei80b]). The level 3 description, using traces, is new, to our knowledge, in the field of (nonmonotonic) reasoning (with the possible exception of step-logic, see Section 5.8). Trace semantics for processes in general (not necessarily *reasoning* processes) is of course known from process algebra (see for instance [BW90]), and the temporal semantics of programs (see for instance [Kro87]) can also be seen as such, if we view a (linear) temporal model as a trace. This perspective will play a role in later chapters. There are also logics for modeling (database) updates (such as transaction logic, see [BK93, BK95]) or the behavior of object-oriented systems (such as Troll, see [JSHS96]) with a semantics based on traces. These two approaches (and some others) are described and illustrated in [EEF$^+$98].

Acknowledgments

The formalization of reasoning presented in this chapter, is a generalization of work reported in [EHT95], where the information state frame is fixed (basically, the information state frame of epistemic states of Definition 2.5).

Chapter 3

Specification of Reasoning

Reasoning can be formalized semantically on different levels of abstraction by final belief state operators, multiple belief state operators, and reasoning frame operators. In this section, a number of well-known *specification languages* for these (mathematical) objects will be described. A specification language consists of a (formal) language, together with a semantics, that is, a way of assigning a mathematical structure to (sets of) sentences of the language. If these mathematical structures are, for instance, reasoning traces, then a theory (set of sentences) in the specification language *specifies* (a set of) reasoning traces. The language can then be called a *specification language* for level 3. The use of a suitable specification language is that it gives a precise, insightful and succinct way of identifying a set of (mathematical) objects. To get back to an example we gave earlier, the formula $p \vee q$ is a precise, insightful and succinct way of identifying the set of all valuations in which either p or q is true, or both. We shall start by showing that default logic is a specification language for levels 1, 2 and 3.

3.1 Default logic

Reiter's default logic ([Rei80b], see also [Łuk90], [Eth87]) is a nonmonotonic logic which aims at formalizing defeasible reasoning. The basic idea is that an agent has two kinds of knowledge. Firstly, it has knowledge (about the world) of which it is certain: this knowledge consists of facts and general rules which are certainly true, called the *axioms*. Secondly, it has a number of default rules: rules whose conclusion is not certain, given the premise, but usually, or normally true. The generic example of such a rule is that birds normally fly. The defeasibility of this rule lies in the fact that not all birds fly: the generic (but not normal) exception to this rule is the penguin Tweety, who does not fly.

Let $\mathcal{L}$ be a propositional language (we will restrict ourselves to the propositional case, although the definitions in [Rei80b] are for first-order predicate logic). A *default* *rule* is an expression

$$(\alpha : \beta_1, \ldots, \beta_n / \gamma)$$

where α, β_1 through β_n and γ are propositional formulae of $\mathcal{L}$. The formula α is called the *prerequisite* of the rule, $\beta_1, \ldots, \beta_n$ are the *justifications*, and γ is the *conclusion*. The intended reading of this expression is:

> *if you believe α, and β_1 through β_n are consistent with what you believe, then you can conclude γ.*

The complexity of default logic (both computationally and conceptually) is caused by the justifications: the β_i may be consistent with what you believe *now*, but may become inconsistent *later* (when adding other default conclusions). So, in order to determine if a rule may be applied now, it is necessary to know which other conclusions will be added. This phenomenon determines the necessity of the fixpoint construction in the definition below.

An agent performing default reasoning has axioms and default rules. Formally, a *default theory* is a pair $\langle D, W\rangle$, where D is a set of default rules, and W is a set of propositional formulae. Using the default rules, the agent extends W to a set of conclusions (formulae), called an *extension*. This can in general be done in multiple ways. The definition of an extension below is not Reiter's original definition, but a (slight) variant of a definition shown to be equivalent in [Rei80b].

Definition 3.1 (Extension of a default theory) Let $\langle D, W\rangle$ be a default theory.

1. A set of sentences E is an *extension* of $\langle D, W\rangle$ if $E = \bigcup_{i=0}^{\infty} E_i$ where the sets E_i are defined as follows:

$$\begin{array}{lcl} E_0 & = & Cn(W), \text{ and for } i \geq 0\text{:} \\ E_{i+1} & = & Cn(E_i \cup \{\gamma \mid (\alpha, \beta_1, \ldots, \beta_n)/\gamma \in D, \alpha \in E_i \\ & & \qquad\qquad \text{and } \neg\beta_j \notin E \text{ for } 1 \leq j \leq n\}). \end{array}$$

2. The set of extensions of $\langle D, W\rangle$ is denoted by $\mathsf{Ext}(D, W)$.

3. If a formula φ occurs in all extensions of $\langle D, W\rangle$, it is called a *sceptical consequence* of $\langle D, W\rangle$. If φ occurs in at least one extension, it is called a *credulous consequence*.

Note the dependence of the sets E_i on the set E. It is straightforward to check that for an extension E, it holds that $E = Cn(E)$, $E_i = Cn(E_i)$, and $E_i \subseteq E_{i+1}$ for $i \in \mathbb{N}$.

Definition 3.2 (Trace of an extension) For an extension E, let the sets E_i be defined as in Definition 3.1. The sequence $\{Mod(E_i)\}_{i\in\mathbb{N}}$ is called the *trace* of E, and is denoted as $\mathsf{Tr}(E)$.

The name 'trace' of an extension is of course not a coincidence: given the information states IS^{ep} of Definition 2.5, a sequence $\{Mod(E_i)\}_{i\in\mathbb{N}}$ is a reasoning trace. Now let a set of defaults D be fixed. For every set of propositional formulae X, one can consider the default theory $\langle D, X\rangle$. Now define an operator $\mathsf{Tr}_D : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathsf{Traces}(\mathsf{IS}^{ep}))$ by

$$\mathsf{Tr}_D(X) = \{\mathsf{Tr}(E) \mid E \in \mathsf{Ext}(D, X)\}.$$

It is easy to check that this is a reasoning frame operator. Viewed this way, default logic (specifically, sets of defaults) offers a specification language for reasoning frame operators. The reasoning frame operators thus specified enjoy a number of properties (which are straightforward to prove):

Proposition 3.3 (Properties of default logic reasoning frame operators) Let D be a set of default rules.

1. For any $X \subseteq \mathcal{L}$ and $E \in \mathsf{Ext}(D, X)$, the trace of E is conservative and eager.
2. The reasoning frame operator Tr_D satisfies non-inclusiveness, uniqueness of traces and invariance.

Another property of the trace $\mathsf{Tr}(E)$ of an extension E, is that $\lim\ \mathsf{Tr}(E) = Mod(E)$. Let us recall the definition of the operator Γ_D^{dl} from Example 2.12: $\Gamma_D^{dl}(X) = \{Mod(E) \mid E \in \mathsf{Ext}(D, X)\}$. This operator is the MBSO associated with Tr_D, i.e., $\Gamma_{\mathsf{Tr}_D} = \Gamma_D^{dl}$. This means that default logic can also be viewed as a specification language for MBSOs. In fact, Reiter's original definition did not use traces, but a fixpoint operator.

Proposition 3.4 (Reiter's original definition) Let $\langle D, W\rangle$ be a default theory. Define an operator $\Delta : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathcal{L})$ as follows. For $S \subseteq \mathcal{L}$, $\Delta(S)$ is the smallest set satisfying:

- D1 $W \subseteq \Delta(S)$;
- D2 $\Delta(S) = Cn(\Delta(S))$;
- D3 If $(\alpha, \beta_1, \dots, \beta_n)/\gamma \in D$, $\alpha \in \Delta(S)$ and $\neg\beta_1, \dots, \neg\beta_n \notin S$, then $\gamma \in \Delta(S)$.

A set of formulae E is an extension if and only if $\Delta(E) = E$.

Proof: See [Rei80b]. □

As the operator Δ depends on the default theory $\langle D, W\rangle$, it can be denoted by $\Delta_{\langle D,W\rangle}$. Then the operator $\mathcal{B}_D^{dl}$ (of Example 2.12) can also be defined by $\mathcal{B}_D^{dl}(X) = \{E \subseteq \mathcal{L} \mid \Delta_{\langle D,X\rangle}(E) = E\}$. It is easily seen that the FBSO $\Phi_{\mathcal{B}_D^{dl}}$ associated with $\mathcal{B}_D^{dl}$ gives the sceptical conclusions:

$$\Phi_{\mathcal{B}_D^{dl}}(X) = \mathrm{glb}(\mathcal{B}_D^{dl}(X)) = \bigcap\{E \mid E \in \mathcal{B}_D^{dl}(X)\} = \bigcap\{E \mid E \in \mathsf{Ext}(D, X)\}.$$

A set of defaults thus specifies an FBSO $\Phi_{\mathcal{B}_D^{dl}}$. As default logic can be used to specify reasoning frame operators, MBSOs and FBSOs, it offers a specification language for reasoning processes at levels 1, 2 and 3.

3.2 Logic programming

The field of logic programming originally studied subsets of predicate logic that can be executed ([Kow74]). Gradually, with the introduction of negation as failure, it became part of the fields of knowledge representation and nonmonotonic reasoning. One of the main impetuses of the field was the development of efficient implementations of logic programming languages, like PROLOG. We will give a brief introduction to positive logic programming; other variants will be discussed later (in Section 5.3). We will use the information state frame of two-valued states of Definition 2.2.

Definition 3.5 (Positive logic program) Let P be a set of propositional atoms. A *positive logic program* is a set of *rules* of the form $p_0 \leftarrow p_1, \ldots, p_n$, where $p_i \in P$ for $0 \leq i \leq n$. When $n = 0$, $p_0 \leftarrow$ is called a *fact*.

A rule $p_0 \leftarrow p_1, \ldots, p_n$ has the following meaning: if the reasoning agent has derived the atoms $p_1, \ldots, p_n$, then it can also derive p_0. The intuitive semantics of a program is that an atom is true if it can be derived from the program, and false otherwise (this is the negation as failure).

Definition 3.6 (Immediate consequence operator) Let Q be a positive logic program. Define the immediate consequence operator $T_Q : \mathsf{IS} \to \mathsf{IS}$ by:

$$T_Q(m)(p) = \begin{cases} 1 & \text{if } m(p) = 1 \text{ or if there is a rule } p \leftarrow p_1, \ldots, p_n \in Q \\ & \quad \text{such that } m(p_i) = 1 \text{ for } 1 \leq i \leq n \text{ ;} \\ 0 & \text{otherwise} \end{cases}$$

The immediate consequence operator defines what the agent can derive in one reasoning step. A reasoning trace associated with a logic program thus arises naturally as the repeated application of this operator.

Definition 3.7 (Reasoning frame operator of a logic program) Let Q be a positive logic program. For a valuation $n \in \mathsf{IS}^{2val}$, define the sequence $tr_Q(n) =$

$\{m_i\}_{i\in\mathbb{N}}$ by

$$\begin{aligned} m_0 &= n \text{ , and for } i \geq 0 : \\ m_{i+1} &= T_Q(m_i). \end{aligned}$$

Given a set of inputs $X \subseteq P$, let m_X be the valuation defined by: $m_X(p) = 1$ iff $p \in X$. Now define the reasoning frame operator $\mathcal{T}_Q$ by

$$\mathcal{T}_Q(X) = \{tr_Q(m_X)\}.$$

It is well-known that a trace of a logic program is conservative and eager. The reasoning frame operator of a positive logic program is very well-behaved. For every set of input facts, there is a unique trace, so the reasoning frame satisfies non-inclusiveness and uniqueness of traces. The associated MBSO, $\Gamma_{\mathcal{T}_Q}$, is simple: $\Gamma_{\mathcal{T}_Q}(X) = \{\lim tr_Q(m_X)\}$, and its associated FBSO, which we will denote by Φ to avoid nested subscripts, is given by $\Phi(X) = \lim tr_Q(m_X)$. The valuation $\lim tr_Q(m_X)$ is what is classically considered to be the semantics of a program $Q \cup X$ (where an atom $p \in X$ is identified with a fact $p \leftarrow$). There is also a definition of this semantics that does not use traces (or the immediate consequence operator).

Proposition 3.8 (Minimal model of a positive program) For a positive logic program Q, consider the set $A(Q) \subseteq \mathsf{IS}$, defined as

$$A(Q) = \{m \in \mathsf{IS} \mid m \models p_1 \wedge \ldots \wedge p_n \rightarrow p_0 \text{ for every rule } p_0 \leftarrow p_1, \ldots, p_n \in Q\}.$$

Then $A(Q)$ has a minimum (with respect to $(\mathsf{IS}, \preceq)$), denoted by $\mathit{Minmod}(Q)$, and it holds:

$$\mathit{Minmod}(Q) = \lim tr_Q(m_\emptyset).$$

This means that we can also directly define $\Phi(X) = \mathit{Minmod}(Q \cup X)$.

Logic programming (with positive programs) offers a specification language for reasoning on levels 1, 2 and 3, although it is a language with restricted expressiveness: for level 2, for instance, only MBSOs containing a single information state can be defined. Many extensions of positive logic programming exist which allow a set of input facts to have more than one trace. These extensions invariably allow default negation in a rule, and we will consider some extensions later on (Section 5.3).

3.3 Temporal logic

A reasoning process performed by an agent was assumed to be a stepwise process. The following is a simple but important observation:

The steps in a reasoning process are steps in time.

In this view, the agent starts to reason at the *first point in time*, with a set of initial beliefs, and applies its reasoning mechanism, going to the *next point in time*, again applies its reasoning mechanism to the information state it has *at that point in time*, again going to a *next point in time*, and so on. A reasoning trace can thus be regarded as a *temporal model*, where the natural numbers are the mathematical formalization of time. (Of course it is a *choice* to model time in this way; a choice which is often made in temporal specification of processes — see for example [Eme90] — and which will turn out to allow many interesting phenomena to be described. However, different models of time — see for example [Ben91b] —, for example infinite towards the past or dense, could also make sense and might be required for describing some relevant features of reasoning processes.) A reasoning frame is then a set of temporal models. This temporal element is implicitly present in default logic and logic programming, but we want to make this explicit. A natural candidate for a language to specify sets of temporal models, is (some variant of) temporal logic.

The idea of using temporal logic to specify processes, is of course not new. In theoretical computer science, many temporal logics for specifying and reasoning about the behavior of processes have been proposed and studied (see for instance [Eme90, MP92]). However, the behavior specified in these temporal logics, is not reasoning behavior but the behavior of (hardware) processors in a computer system. For instance, a state of a processor (at a certain point in time) is very different from a mental (information) state of a reasoning agent (which generally contains uncertain and partial information). Therefore, new temporal logics should be introduced and studied which are suited in particular for describing the behavior over time of a reasoning agent. The next chapter describes a number of possibilities for such a temporal logic. Properties and suitability of these logics are studied in later chapters.

3.4 Conclusions and related work

We have shown that both default logic and logic programming can be seen as specification languages for the mathematical formalizations of reasoning of Chapter 2, on the first three levels of abstraction. The definition of the trace of an extension, although not called trace, is already in [Rei80b], and the notion of the immediate consequence operator being applied iteratively to the empty set (generating a trace), is also known (see for example [Prz90]). However, the interpretation of both frameworks as specification languages on level 3 (and maybe also on level 2), is novel to our knowledge.

Acknowledgments

Section 3.1 is based on [EHT95], and Section 3.3 is based on [ET96c].

Chapter 4

Temporal Logics of Information

Temporal logic can be used to specify processes, and to reason about their properties. The idea of using temporal logic as a specification language was briefly introduced in the previous chapter. In this chapter, a number of temporal logics will be introduced that are specifically geared towards *reasoning* processes. After defining a number of suitable temporal logics, at the end of this chapter a general definition is given of the reasoning frame operator defined by a temporal theory in one of these logics. Chapter 5 describes a number of forms of reasoning that can be specified in one of these logics.

When designing a logic capable of describing the behavior of reasoning processes over time, at least two important decisions have to be made: what is a state in a reasoning process, and which formalization of time is suited best for the purpose? To start with the first question, a state should be an information state, and we have seen a number of possibilities in Chapter 2. In order for a temporal logic to be a specification language for level 3, it should be able to describe traces. One obvious choice for the formalization of time, is to use linear discrete time with a starting point, structures isomorphic to the natural numbers (or even the natural numbers themselves). But another possibility is to use branching time structures. One branch in such a structure also defines a reasoning trace. In theoretical computer science there has been much debate whether time should be modeled as linear or branching (towards the future) (e.g., see [BRR89]). The most important differences between these two approaches are that linear time logics have in general a lower complexity but also less expressivity than the corresponding branching time logics.

Given these choices to be made on the nature of states and time, many different temporal logics can be defined. Rather than setting up a general framework in which these choices are parameters, we will describe a number of concrete temporal logics. Logics based on a different choice of these parameters, can be defined by analogy.

4.1 Temporal epistemic logic

In an information state, the agent has certain knowledge or belief. Epistemic modal logics are intended to describe knowledge and belief of rational agents. The temporal logic presented in this section, temporal epistemic logic (TEL), uses the epistemic modal logic S5 to describe the information states of the agent. There has been much discussion about the use of S5 for capturing the knowledge of an agent. We will take the same position as taken by [FHMV95], namely that the appropriate formalization of knowledge depends on the 'application'. For many of our purposes, S5 is suitable for playing that role and will be used, but for other purposes (such as the temporal view on proof systems in Section 5.4), other logics will be used (such as partial logic, see Section 4.2).

The temporal epistemic logic introduced in this section is meant to *describe* the temporal epistemic behavior of the agent as seen from the outside, it is not the logic to be used by the agent itself. (This use of logic is called *external* in [FHMV95]; see also the extensive discussion of these two uses of logic in [Lev90], where our use is called *objective*. To quote from that paper: "Thus, logic is being used here as a specification tool to *describe* a reasoner rather than as a calculus to be *used* by one." TEL's intended use is exactly as such a "specification tool".)

As TEL is defined 'on top of' S5 (TEL is the result of *temporalizing* S5, in the terminology of [FG92]), we will first give a brief introduction to S5.

4.1.1 S5

Propositional logic will be taken as the basic logic in which the agent can describe its knowledge. A modal operator K is used to denote the agent's knowledge. The agent may perform both positive and negative introspection.

Definition 4.1 (Epistemic language) Let P be a (finite or countably infinite) set of propositional atoms. The language $\mathcal{L}_{S5}$ is the smallest set closed under:

- if $p \in P$ then $p \in \mathcal{L}_{S5}$;
- if $\varphi, \psi \in \mathcal{L}_{S5}$ then $K\varphi, \varphi \wedge \psi, \neg\varphi \in \mathcal{L}_{S5}$.

Furthermore, we introduce the following abbreviations:

$$\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi), \varphi \to \psi \equiv \neg\varphi \vee \psi, M\varphi \equiv \neg K \neg\varphi, \top \equiv p \vee \neg p, \bot \equiv \neg\top.$$

If every atom occurring in a formula φ is in the scope of a K operator, we call φ *subjective*.

An example of a subjective formula is $\neg Kp \wedge K(q \to p)$, whereas $K(p \wedge q) \vee s$ is not subjective. In the rest of this chapter we will only be interested in subjective formulae since they describe (just) the knowledge and ignorance of the agent. On this subset of subjective formulae, S5 coincides with the logic KD45, which is sometimes considered to be *the* logic of belief (the extra axiom of S5, $K\varphi \to \varphi$, is valid in KD45 whenever φ is subjective).

In the usual S5 semantics, a model is a triple (W, R, π) where W is a set of worlds, R is an equivalence relation on W and π is a function that assigns a propositional valuation to each world in W. In the case of one agent, however, we may restrict ourselves to *normal* S5-models, in which the relation is universal (each world is accessible from every other world), and worlds are identified with propositional valuations (see [MH95] for a proof of soundness and completeness of S5 with respect to these semantics). This is no longer true in the case of more than one agent.

Definition 4.2 (S5 semantics) A propositional valuation of signature P is a function from P into $\{0,1\}$ where 0 stands for false and 1 for true. The set of such valuations will be denoted by $\mathrm{Val}(P)$. A *normal S5-model* M is a non-empty set of valuations. The truth of an S5-formula φ in such a model, evaluated in a world $m \in M$, denoted $(M,m) \models_{S5} \varphi$, is defined inductively:

$$\begin{array}{lcl} (M,m) \models_{S5} p & \Leftrightarrow & m(p) = 1, \text{ for } p \in P \\ (M,m) \models_{S5} \varphi \wedge \psi & \Leftrightarrow & (M,m) \models_{S5} \varphi \text{ and } (M,m) \models_{S5} \psi \\ (M,m) \models_{S5} \neg\varphi & \Leftrightarrow & \text{it is not the case that } (M,m) \models_{S5} \varphi \\ (M,m) \models_{S5} K\varphi & \Leftrightarrow & (M,m') \models_{S5} \varphi \text{ for every } m' \in M \end{array}$$

We have the following elementary results for subjective formulae:

Proposition 4.3 (Subjective formulae)

1. Let φ be a subjective formula. For a normal S5-model M and $m_1, m_2 \in M$ it holds:

   $$(M, m_1) \models_{S5} \varphi \Leftrightarrow (M, m_2) \models_{S5} \varphi.$$

   We define $M \models_{S5} \varphi$ iff $(M,m) \models_{S5} \varphi$ for some, or, equivalently, all $m \in M$.

2. An S5-formula φ is subjective if and only if it is equivalent to a formula of the form $K\varphi$ with $\varphi \in \mathcal{L}_{S5}$.

3. For any propositional formula α and S5-models M, N:

   $$(M \models_{S5} K\alpha \ \&\ M \preceq N) \Rightarrow N \models_{S5} K\alpha$$

   where $\preceq$ is the information order on IS^{ep}.

Proof: Straightforward. $\square$

Note that a normal S5-model is the same as an epistemic state from IS^{ep} of Definition 2.5, with the exception of the empty set, which is not an S5-model. Whenever we talk about S5-models in the rest of this thesis, we will mean normal S5-models.

4.1.2 Temporalizing S5

In S5, we can talk about the knowledge (or belief) of the agent at a fixed point in time. In order to describe past and future of the agent's knowledge, we introduce temporal operators P, H, F, G, Y, X and $\Box$, standing for "sometimes in the past", "always in the past", "sometimes in the future", "always in the future", "at the previous point in time", "at the next point in time" and "always" respectively. Note that we do not want to talk about the agent's knowledge of the future and past, but about the future and past of the agent's knowledge. Therefore temporal operators need never occur within the scope of the epistemic K operator. This is reflected in the definition of the temporal epistemic language.

Definition 4.4 (Temporal epistemic language) The language $\mathcal{L}_{TEL}$ is the smallest set closed under

- if $\varphi \in \mathcal{L}_{S5}$ is subjective, then $\varphi \in \mathcal{L}_{TEL}$;
- if $\alpha, \beta \in \mathcal{L}_{TEL}$ then $\alpha \wedge \beta, \neg\alpha, P\alpha, F\alpha, Y\alpha, X\alpha \in \mathcal{L}_{TEL}$.

Again, the abbreviations for $\vee$, $\rightarrow$, $\top$ and $\bot$ are introduced, as well as:

$$G\alpha \equiv \neg F \neg \alpha, H\alpha \equiv \neg P \neg \alpha \text{ and } \Box\alpha \equiv H\alpha \wedge \alpha \wedge G\alpha.$$

Based on the set of natural numbers as the flow of time and the notion of normal S5-model as formalization of a state, the following semantics is introduced for temporal epistemic logic (TEL):

Definition 4.5 (Semantics of TEL) A TEL-model $\mathcal{M}$ is a sequence $(\mathcal{M}_t)_{t\in\mathbb{N}}$ of normal S5-models. The truth of a formula $\varphi \in \mathcal{L}_{TEL}$ in $\mathcal{M}$ at time point $t \in \mathbb{N}$, denoted $(\mathcal{M}, t) \models \varphi$, is defined inductively as follows:

$$\begin{array}{lcl}
(\mathcal{M},t) \models \varphi & \Leftrightarrow & \mathcal{M}_t \models_{S5} \varphi, \text{ if } \varphi \in \mathcal{L}_{S5} \\
(\mathcal{M},t) \models \varphi \wedge \psi & \Leftrightarrow & (\mathcal{M},t) \models \varphi \text{ and } (\mathcal{M},t) \models \psi \\
(\mathcal{M},t) \models \neg\varphi & \Leftrightarrow & \text{it is not the case that } (\mathcal{M},t) \models \varphi \\
(\mathcal{M},t) \models P\varphi & \Leftrightarrow & \exists s \in \mathbb{N} \text{ such that } s < t \text{ and } (\mathcal{M},s) \models \varphi \\
(\mathcal{M},t) \models F\varphi & \Leftrightarrow & \exists s \in \mathbb{N} \text{ such that } t < s \text{ and } (\mathcal{M},s) \models \varphi \\
(\mathcal{M},t) \models Y\varphi & \Leftrightarrow & t > 0 \text{ and } (\mathcal{M},t-1) \models \varphi \\
(\mathcal{M},t) \models X\varphi & \Leftrightarrow & (\mathcal{M},t+1) \models \varphi
\end{array}$$

A formula φ is true in a model $\mathcal{M}$, denoted $\mathcal{M} \models \varphi$, if for all $t \in \mathbb{N}, (\mathcal{M}, t) \models \varphi$. A set of formulae T is true in a model $\mathcal{M}$, denoted $\mathcal{M} \models T$, if $\mathcal{M} \models \varphi$ for all $\varphi \in T$. If φ is true in all models we write $\models \varphi$ (φ is *valid*), and we write $T \models \varphi$ (φ is a *semantical consequence* of T) if for all models $\mathcal{M}$ and $t \in \mathbb{N}, (\mathcal{M}, t) \models T$ implies $(\mathcal{M}, t) \models \varphi$. We will sometimes write $\mathcal{M}(t)$ for $\mathcal{M}_t$.

In Section 9.1, TEL is studied in more detail. In the next section we will impose an extra restriction on our models.

4.1.3 Conservativity

We want to use temporal formulae for describing the behavior of a reasoning agent. The reasoning will be assumed to be *conservative*, that is, the agent's knowledge increases as it is reasoning. Although the actual implementation of the reasoning behavior may involve backtracking or the addition of extra assumptions which may later be retracted, we are interested only in the increase of knowledge over time. This presupposes a world which does not change.

It is easy to see that a temporal model is just a trace in the information state frame $\langle \mathcal{L}, Cn, \mathsf{IS}^{ep}, \preceq, \mathrm{Th}\rangle$ of Definition 2.5. This provides us with a definition of conservativity of temporal models.

Definition 4.6 (Conservative models — TELC)

1. A TEL-model $\mathcal{M}$ is called *conservative* if it is conservative as a trace in the information state frame of Definition 2.5, that is, if

$$\mathcal{M}_s \preceq \mathcal{M}_{s+1} \quad \forall s \in \mathbb{N}.$$

2. Validity and semantical consequence restricted to the class of conservative models *(TELC-models)* is denoted by $\models^c$. The resulting logic will be called TELC.

Note that for any conservative model $\mathcal{M}$, time point $s \in \mathbb{N}$ and propositional formula φ: if $(\mathcal{M}, s) \models K\varphi$, then for $t > s$ also $(\mathcal{M}, t) \models K\varphi$. This means that whenever a propositional formula is known, it will remain known in the future. This is of course not the case in general for non-propositional formulae. If we define $\mathcal{M}$ by $\mathcal{M}_0 = \mathrm{Val}(P)$ and $\mathcal{M}_i = \mathit{Mod}(\{p\})$ for $i > 0$ (where $\mathit{Mod}(\{p\})$ is the set of propositional valuations in which p is true), then $(\mathcal{M}, 0) \models M\neg p$ and $(\mathcal{M}, 0) \models KM\neg p$ but $(\mathcal{M}, 1) \models Kp$ so $(\mathcal{M}, 1) \not\models M\neg p$ and $(\mathcal{M}, 1) \not\models KM\neg p$ (and even $(\mathcal{M}, 1) \models K\neg M\neg p$).

The notion of a limit model of a conservative temporal epistemic model is defined according to Definition 2.20.

Definition 4.7 (Limit model of a TELC-model) Let $\mathcal{M}$ be a TELC-model. The limit of $\mathcal{M}$ is defined by:

$$\lim \mathcal{M} = \bigcap \{\mathcal{M}_s \mid s \in \mathbb{N}\}.$$

The limit model of a TELC-model should reflect all the conclusions the agent has drawn over time (and nothing more). When the set of atoms P is infinite, however, it may be the case that although $\mathcal{M}_i$ is non-empty for all $i \in \mathbb{N}$, the limit is empty (and, therefore, not an S5-model). A special condition on temporal models eliminates this possibility.

Definition 4.8 (Closed model)

1. An S5-model M is *closed* if there is a consistent set of propositional formulae T such that
$$M = \{m \in \mathrm{Val}(P) \mid m \models \varphi \text{ for all } \varphi \in T\}.$$
2. A TELC-model $\mathcal{M}$ is *closed* if $\mathcal{M}_s$ is closed for all $i \in \mathbb{N}$.

The term 'closed' is not arbitrary. If we define an operator $Cl : \mathcal{P}(\mathrm{Val}(P)) \to \mathcal{P}(\mathrm{Val}(P))$ by
$$Cl(M) = Mod(Th(M))$$
where for a set A of propositional formulae, $Mod(A) = \{m \in \mathrm{Val}(P) \mid m \models \varphi \text{ for all } \varphi \in A\}$ and $Th(M) = \{\varphi \mid \varphi \text{ a propositional formula such that } m \models \varphi \text{ for all } m \in M\}$ then Cl is a (Kuratowski) closure operator in the topological sense (see for instance [War72]) and a model M is closed if and only if $Cl(M) = M$. Furthermore, one can define a metric on propositional valuations. Fix an enumeration $p_0, p_1, \ldots$ of P, and define $d : \mathrm{Val}(P) \times \mathrm{Val}(P) \to \mathbb{R}$ by
$$d(m,n) = \begin{cases} 0 & \text{if } m = n \\ 2^{-j} & \text{if } j \text{ is the smallest index such that } m(p_j) \neq n(p_j). \end{cases}$$
Then d is a metric on $\mathrm{Val}(P)$ and the closed S5-models are exactly the sets which are closed with respect to the topology induced by d.

It is easy to see that in case P is finite, all S5-models are closed: for an S5-model M, the set of propositional formulae T required in Definition 4.8 can be defined as follows (it is even a singleton):
$$T = \{\bigvee\{\bigwedge\{p \ : \ m \models p\} \wedge \bigwedge\{\neg p \ : \ m \models \neg p\} \mid m \in M\}\}.$$

Proposition 4.9 Let $\mathcal{M}$ be a closed TELC-model. Then $\lim\ \mathcal{M}$ is an S5-model, and for all propositional formulae φ, and $s \in \mathbb{N}$:
$$(\mathcal{M}, s) \models FK\varphi \Leftrightarrow \mathcal{M} \models FK\varphi \Leftrightarrow \lim\ \mathcal{M} \models_{S5} K\varphi$$
or, equivalently,
$$Th(\lim\ \mathcal{M}) = \bigcup_{s=0}^{\infty} Th(\mathcal{M}_s).$$

Proof: The equivalence $(\mathcal{M}, s) \models FK\varphi \Leftrightarrow \mathcal{M} \models FK\varphi$ follows straightforwardly from conservativity and the definition of the F-operator, and does not depend on closedness. Now suppose $\mathcal{M}$ is a closed TELC-model. This means that for every $s \in \mathbb{N}$ there must exist consistent sets T_s of propositional formulae, such that

$\mathcal{M}_s = Mod(T_s)$. Without loss of generality, these sets T_s can be taken closed under propositional consequence. As $\mathcal{M}$ is conservative, it must hold that $T_0 \subseteq T_1 \subseteq \ldots$. Then $\bigcup_{s=0}^{\infty} T_s$ is closed under propositional provability, and we have:

$$\begin{array}{rcll} Th(\lim\ \mathcal{M}) & = & Th(\bigcap_{s=0}^{\infty} \mathcal{M}_s) & \text{(by definition of lim)} \\ & = & Th(\bigcap_{s=0}^{\infty} Mod(T_s)) & \text{(by assumption)} \\ & = & Th(Mod(\bigcup_{s=0}^{\infty} T_s)) & \\ & = & Cn(\bigcup_{s=0}^{\infty} T_s) & \\ & = & \bigcup_{s=0}^{\infty} T_s & \text{(as } \bigcup_{s=0}^{\infty} T_s \text{ is closed under provability)} \\ & = & \bigcup_{s=0}^{\infty} Th(\mathcal{M}_s) & \text{(by definition of } T_s \text{ and its closure under} \\ & & & \text{provability).} \end{array}$$

It is easy to see that this implies that for all propositional formulae φ we have $\mathcal{M} \models FK\varphi \Leftrightarrow \lim\ \mathcal{M} \models_{S5} K\varphi$. To show that $\lim\ \mathcal{M}$ is an S5-model (i.e., that it is non-empty), we only have to show that $Th(\lim\ \mathcal{M})$ is consistent in propositional logic. This is equivalent to consistency of $\bigcup_{s=0}^{\infty} Th(\mathcal{M}_s)$. If we can deduce falsity from $\bigcup_{s=0}^{\infty} Th(\mathcal{M}_s)$, then by compactness, we can deduce it from a finite subset, which must be contained in $Th(\mathcal{M}_s)$ for some $s \in \mathbb{N}$. But this contradicts the fact that $\mathcal{M}_s$ is an S5-model. □

4.2 Temporal partial logic

In the previous section, we introduced a temporal logic based on S5-models as information states. The variant of this logic using three-valued models as information states is the subject of this section. So, we assume the information state frame $\langle \mathcal{L}, C_{\mathcal{L}}, \mathsf{IS}^{3val}, \preceq, \mathrm{Th} \rangle$ of Definition 2.3.

Partial propositional logic can be temporalized in a way analogous to S5. In the case of S5, there is an operator (K) referring to the knowledge of the agent, and we only considered subjective formulae, which are essentially boolean combinations of formulae $K\varphi$ with $\varphi \in \mathcal{L}_{S5}$. For partial logic, we also introduce an operator,

C, denoting the agent's knowledge. For a propositional formula φ, the formula $C\varphi$ means that the agent knows φ (this C-operator should not be confused with the operator for 'common knowledge' in epistemic logic with more than one agent, see [FHMV95]).

Definition 4.10 (Temporal partial language) Let P be a (finite or countably infinite) set of propositional atoms. The language $\mathcal{L}_{TPL}$ is the smallest set closed under:

- if φ is a propositional formula then $C\varphi \in \mathcal{L}_{TPL}$;
- if $\alpha, \beta \in \mathcal{L}_{TPL}$ then $\alpha \wedge \beta$, $\neg\alpha$, $P\alpha$, $F\alpha$, $X\alpha$, $Y\alpha \in \mathcal{L}_{TPL}$.

Again the abbreviations for $\vee$, $\rightarrow$, $\top$, $\bot$, H, G and $\Box$ are introduced (see Definition 4.4).

The semantics of TPL are analogous to TEL.

Definition 4.11 (Semantics of TPL) A TPL-model is a sequence $(\mathcal{M}_t)_{t\in\mathbb{N}}$ where $\mathcal{M}_t \in \mathsf{IS}^{3val}$ for all $t \in \mathbb{N}$. We will often write $\mathcal{M}(t)$ for $\mathcal{M}_t$. The truth of a formula $\varphi \in \mathcal{L}_{TPL}$ in $\mathcal{M}$ at time point $t \in \mathbb{N}$, denoted $(\mathcal{M}, t) \models \varphi$, is defined inductively as follows:

$$
\begin{array}{lcl}
(\mathcal{M},t) \models C\alpha & \Leftrightarrow & \mathcal{M}_t(\alpha) = 1, \text{ if } \alpha \text{ is a propositional formula} \\
(\mathcal{M},t) \models \varphi \wedge \psi & \Leftrightarrow & (\mathcal{M},t) \models \varphi \text{ and } (\mathcal{M},t) \models \psi \\
(\mathcal{M},t) \models \neg\varphi & \Leftrightarrow & \text{it is not the case that } (\mathcal{M},t) \models \varphi \\
(\mathcal{M},t) \models P\varphi & \Leftrightarrow & \exists s \in \mathbb{N} \text{ such that } s < t \text{ and } (\mathcal{M},s) \models \varphi \\
(\mathcal{M},t) \models F\varphi & \Leftrightarrow & \exists s \in \mathbb{N} \text{ such that } t < s \text{ and } (\mathcal{M},s) \models \varphi \\
(\mathcal{M},t) \models Y\varphi & \Leftrightarrow & t > 0 \text{ and } (\mathcal{M},t-1) \models \varphi \\
(\mathcal{M},t) \models X\varphi & \Leftrightarrow & (\mathcal{M},t+1) \models \varphi
\end{array}
$$

A formula φ is true in a model $\mathcal{M}$, denoted $\mathcal{M} \models \varphi$, if for all $t \in \mathbb{N}, (\mathcal{M}, t) \models \varphi$. For a set of formulae T, $(\mathcal{M}, t) \models T$ means that $(\mathcal{M}, t) \models \varphi$ for all $\varphi \in T$. If φ is true in all models we write $\models \varphi$ (φ is *valid*), and for a set of formulae T we write $T \models \varphi$ (φ is a *semantical consequence* of T) if for all models $\mathcal{M}$ and $t \in \mathbb{N}, (\mathcal{M}, t) \models T$ implies $(\mathcal{M}, t) \models \varphi$.

The C-operator allows us to express for instance that a propositional formula α is unknown: $\neg C\alpha \wedge \neg C\neg\alpha$ is true in a partial model exactly when α has the value unknown in the model.

We are again interested in conservative models. A TPL-model $\mathcal{M}$ is conservative if it is conservative as a trace in IS^{3val}, i.e. if $\mathcal{M}_s \preceq \mathcal{M}_{s+1}$ for all $s \in \mathbb{N}$, where $\preceq$ is the information ordering on three-valued models of Definition 2.3.

Definition 4.12 (TPLC) Validity and semantical consequence restricted to the class of conservative models *(TPLC-models)* will be denoted by $\models^c$.

As in the case of conservative TELC-models, the following holds: if $\mathcal{M}$ is a conservative TPL-model, φ is a propositional formula, and for some $s \in \mathbb{N}$ we have $(\mathcal{M}, s) \models C\varphi$, then for all $t > s$ it holds that $(\mathcal{M}, t) \models C\varphi$.

The notion of a limit model of a conservative temporal partial model is again defined according to Definition 2.20.

Definition 4.13 (Limit model of a TPLC-model) Let $\mathcal{M}$ be a TPLC-model. The limit of $\mathcal{M}$ is defined by:

$$\lim\ \mathcal{M}(a) = \begin{cases} 1 & \text{if there exists } t \in \mathbb{N} \text{ such that } \mathcal{M}_t(a) = 1 \\ 0 & \text{if there exists } t \in \mathbb{N} \text{ such that } \mathcal{M}_t(a) = 0 \\ u & \text{otherwise} \end{cases}$$

In temporal partial logic, the limit always contains exactly what has been derived in time: for every propositional formula φ it holds:

$$\begin{array}{lcl} \mathcal{M} \models FC\varphi & \Leftrightarrow & \lim\ \mathcal{M}(\varphi) = 1 \\ \mathcal{M} \models FC\neg\varphi & \Leftrightarrow & \lim\ \mathcal{M}(\varphi) = 0. \end{array}$$

Of course, there are many other temporal logics like TEL (TELC) and TPL (TPLC), using other kinds of information states. In the next section, we will look at branching time logics.

4.3 From linear to branching time

The temporal logics described in the previous two sections were based on linear time. The dynamic behavior of a reasoning agent is formalized by a set of linear time models. During the reasoning, the agent in a sense chooses one of these models. A different way of formalization of the variety of patterns is by branching time temporal models, where each branch represents one of the patterns.

Formalization by a set of linear time models has the advantage of a very simple model structure. But the disadvantage is that the possible choices and the time points at which they should be made are not covered explicitly in the formalization itself. Branching time models represent these choices as points where the flow of time branches. In this section, we will define a branching time logic, parameterized by the logic which is to be temporalized (this could be S5 or partial logic, or any other logic). In the area of temporal specification of processes in theoretical computer science, branching time logics have also been defined (and there has been, or is still, a debate about the relative merits of linear and branching time; we hope this section furthers the understanding of their interrelation somewhat), the most well-known of which are CTL (see [EC82]) and CTL*(see [CES83]; an introduction and comparison of both logics is given in [ES89]).

Given the fact that these approaches are formalizations of (more or less) the same phenomenon, it is natural to study formal connections between them. The branches of a branching time temporal model can be viewed as linear time models. On the other hand, it may be possible to aggregate linear time models into a branching time model. In [ET94a] universal (algebraic) constructions were defined in category-theoretic terms, which enabled the study of these approaches and connections between them. In this thesis, we will not treat this more general case. Instead, the constructions of [ET94a] in the case of temporal models are given and logical properties of these constructions are investigated.

4.3.1 Branching time logic

TEL is the result of temporalizing S5 over the natural numbers. Here, a class of time structures will be defined which describe branching time. Such time structures will be called *flows of time*. The logic that is to be temporalized, is kept abstract: we assume an information state frame (Definition 2.1) is given which supplies us with a language $\mathcal{L}_0$, states (IS) and a way of knowing which formulae are true in a state (Th). We start by defining the flows of time.

4.3.1.1 Flows of time

Definition 4.14 (Flow of time) A (discrete) *flow of time* $(T, <)$ is a pair consisting of a nonempty set T of time points, and a binary relation $<$ on $T \times T$, called the *immediate successor relation* that is irreflexive, antisymmetric and intransitive. For $s, t \in T$ the expression $s < t$ denotes that t is an *immediate successor* of s, and that s is an *immediate predecessor* of t. We denote the (irreflexive) transitive closure of this binary relation $<$ by $\ll$. A flow of time is called *linear* if $\ll$ is a total ordering.

Definition 4.15 (Sub-ft and branch)

1. A flow of time $(T', <')$ is called a *sub-ft (sub-flow of time)* of a flow of time $(T, <)$ if $T' \subseteq T$ and $<' = < \cap (T' \times T')$.

2. A sub-flow of time T' is *right* or *successor* (respectively *left* or *predecessor*) *complete* in T with respect to $t \in T'$ if for all $u \in T$ with $t \ll u$ (respectively $u \ll t$) it holds that $u \in T'$.

3. A *branch* in a flow of time T is a sub-ft $B = (T', <')$ of T such that:

   (a) $\ll' = \ll \cap (T' \times T')$ is a total ordering on $T' \times T'$.

   (b) Every $s \in T'$ with a successor in T also has a successor in T':
   for all $s \in T', t \in T : s < t \Rightarrow$ there is a $t' \in T' : s < t'$.

   (c) Every $t \in T'$ with a predecessor in T also has a predecessor in T':
   for all $s \in T, t \in T' : s < t \Rightarrow$ there is an $s' \in T' : s' < t$.

   (d) For all $s \in T', t \in T, u \in T' : s \ll t \ll u \Rightarrow t \in T'$.

Notice that branches can be viewed as linear temporal models.

Definition 4.16 (Minimal element, root, path)

1. An element t of T is called a *minimal element* if there exists no $s \in T$ with $s < t$. We call t a *root* if for all $u \in T$ it holds that $u = t$ or $t \ll u$.
2. We call T *well-founded* if there do not exist infinite descending chains of elements $s_i < s_{i-1}$.
3. A *(finite) path* is a finite sequence of point $s_0, \ldots, s_n$ such that $s_i < s_{i+1}$ for all $0 \leq i \leq n-1$. We call s_0 the starting point and s_n the endpoint of the path.

We will make additional assumptions on the flow of time: that it describes a discrete tree structure where time branches in the direction of the future, and where time is infinite towards the future.

Definition 4.17 (Tree and forest)

1. The following properties on flows of time are defined:
   (a) *Successor existence*
   Every time point has at least one successor:
   for all $s \in T$ there exists a $t \in T$ such that $s < t$.
   (b) *Rooted*
   A flow of time is rooted if it has a root.
   (c) *Left linear*
   For all t the set of s with $s \ll t$ is totally ordered by $\ll$.
2. A flow of time is called a *tree* if it is rooted and left linear.
3. A flow of time is called a *forest* if it is well-founded and left linear.

Note that a forest is just a disjoint union of trees. From now on we will assume all flows of time to be forests satisfying successor existence.

Observation 4.18 Suppose T is a forest. Every non-minimal element has a unique predecessor. For every non-minimal $t \in T$ there is a unique minimal element m with $m \ll t$ and a unique path with t as end point and m as starting point; this path gives a finite ordered enumeration of $\{s \mid s \ll t\} \cup \{t\}$.

The number of elements in the path from t to its corresponding minimal element, minus one, is called the *depth* of t. Using this depth function the time points of a branch may be identified with $\mathbb{N}$. We will sometimes, if no confusion can arise, use the same character $<$ to denote two different relations on different sets, for example as in $(T, <)$ and $(T', <)$.

4.3.1.2 Temporal models

A temporal model is based on a flow of time, with an information state associated to each point in time.

Definition 4.19 (Temporal model) Let $(T, <)$ be a flow of time. A *temporal model* $\mathfrak{M}$ based on flow of time $(T, <)$ is a triple $(\mathcal{M}, T, <)$, where $\mathcal{M}$ is a mapping

$$\mathcal{M} : T \to \mathsf{IS}.$$

So at any point in time we have an information state describing which knowledge the reasoning process has deduced at that time. We will usually refer to $\mathcal{M}$ as a temporal model based on $(T, <)$. If φ is a formula from $\mathcal{L}_0$, and t is a time point in T, and $\varphi \in \mathrm{Th}(\mathcal{M}(t))$, then we say that in the model $\mathcal{M}$ *at time point t the formula φ is true.*

Definition 4.20 The temporal model $\mathcal{M}'$ is a *submodel* of $\mathcal{M}$ if $(T', <')$ is a subflow of time of $(T, <)$ with $\mathcal{M}(t) = \mathcal{M}'(t)$ for all $t \in T'$. We also call $\mathcal{M}'$ the *restriction* of $\mathcal{M}$ to T', denoted by $\mathcal{M}|_{T'}$. If T' is a branch of T then $\mathcal{M}'$ is called a *branch* of $\mathcal{M}$. For a temporal model $\mathcal{M}$, the set of its branches is denoted by $\mathsf{Br}(\mathcal{M})$.

4.3.1.3 Temporal formulae and their interpretation

We will now define the temporal language $\mathcal{L}_T$ in terms of the language $\mathcal{L}_0$ using temporal operators to describe truth of formulae over time. Because our temporal models based on forests have a more differentiated structure towards the future than towards the past, we will need more operators describing the future than the past. Essentially, we have taken the operators of CTL (see [BPM83]). Also, we do not want any interaction between formulae of $\mathcal{L}_0$ and temporal formulae. Therefore the formulae of $\mathcal{L}_0$ are "shielded" by an operator C (not to be confused with the operator C for partial temporal logic defined earlier).

Definition 4.21 (Temporal language) The temporal language $\mathcal{L}_T$ is defined to be the least set such that:

1. $\varphi \in \mathcal{L}_0 \Rightarrow C\varphi \in \mathcal{L}_T$;
2. $\varphi, \psi \in \mathcal{L}_T \Rightarrow \neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \to \psi \in \mathcal{L}_T$;
3. $\varphi \in \mathcal{L}_T \Rightarrow O\varphi \in \mathcal{L}_T$ (where $O \in \{\exists\mathrm{F}, \forall\mathrm{F}, \exists\mathrm{G}, \forall\mathrm{G}, \exists\mathrm{X}, \forall\mathrm{X}, P, H\}$).

A *temporal theory* is a subset of $\mathcal{L}_T$.

The temporal language is similar to a modal propositional language where the atomic propositions consist of the C operator applied to a formula in $\mathcal{L}_0$. In these definitions, for a temporal model $\mathcal{M}$ based on $(T, <)$, $t \in T$, and $\alpha \in \mathcal{L}_T$, $(\mathcal{M}, t) \models \alpha$ means that α is true in $\mathcal{M}$ at time point t.

Definition 4.22 (Semantics) Let a temporal model $\mathcal{M}$ based on $(T, <)$, and a time point $t \in T$ be given, then inductively define:

1. for $\alpha \in \mathcal{L}_0$: $(\mathcal{M}, t) \models C\alpha \Leftrightarrow \alpha \in \mathrm{Th}(\mathcal{M}(t))$.
2. for $\varphi, \psi \in \mathcal{L}_T$:

$$\begin{array}{lcl} (\mathcal{M}, t) \models \neg\varphi & \Leftrightarrow & \text{it is not the case that } (\mathcal{M}, t) \models \varphi \\ (\mathcal{M}, t) \models \varphi \wedge \psi & \Leftrightarrow & (\mathcal{M}, t) \models \varphi \text{ and } (\mathcal{M}, t) \models \psi. \end{array}$$

3. for $\varphi \in \mathcal{L}_T$:

$$\begin{array}{lcl} (\mathcal{M}, t) \models \exists\mathrm{F}\varphi & \Leftrightarrow & \exists s \in T[t \ll s \ \&\ (\mathcal{M}, s) \models \varphi] \\ (\mathcal{M}, t) \models \exists\mathrm{G}\varphi & \Leftrightarrow & \text{there exists a branch including } t \text{ such that} \\ & & \text{for all } s \text{ in that branch: } [t \ll s \Rightarrow (\mathcal{M}, s) \models \varphi] \\ (\mathcal{M}, t) \models \exists\mathrm{X}\varphi & \Leftrightarrow & \exists s \in T[t < s \ \&\ (\mathcal{M}, s) \models \varphi] \\ (\mathcal{M}, t) \models P\varphi & \Leftrightarrow & \exists s \in T[s \ll t \ \&\ (\mathcal{M}, s) \models \varphi]. \end{array}$$

Furthermore, we introduce the following abbreviations:

$$\begin{array}{rcl} \varphi \vee \psi & \equiv_{\mathrm{def}} & \neg(\neg\varphi \wedge \neg\psi), \\ \varphi \to \psi & \equiv_{\mathrm{def}} & \neg\varphi \vee \psi, \\ \top & \equiv_{\mathrm{def}} & C\alpha \vee \neg C\alpha \text{ (for some } \alpha \in \mathcal{L}_0), \\ \bot & \equiv_{\mathrm{def}} & \neg\top, \\ \forall\mathrm{F}\varphi & \equiv_{\mathrm{def}} & \neg\exists\mathrm{G}(\neg\varphi), \\ \forall\mathrm{G}\varphi & \equiv_{\mathrm{def}} & \neg\exists\mathrm{F}(\neg\varphi), \\ \forall\mathrm{X}\varphi & \equiv_{\mathrm{def}} & \neg\exists\mathrm{X}(\neg\varphi), \text{ and} \\ H\varphi & \equiv_{\mathrm{def}} & \neg P(\neg\varphi). \end{array}$$

For a temporal model $\mathcal{M}$, by $\mathcal{M} \models \varphi$ we mean $(\mathcal{M}, t) \models \varphi$ for all $t \in T$ and by $\mathcal{M} \models K$ we mean $\mathcal{M} \models \varphi$ for all $\varphi \in K$, where K is a set of temporal formulae. The class of temporal models $\mathcal{M}$ for which $\mathcal{M} \models K$ will be denoted by $Mod(K)$.

The property of successor existence can be axiomatized by the formula $\exists\mathrm{F}(\top)$. If in a model $\mathcal{M}$ the formula $P(\top)$ is true at time point t then t must have a predecessor.

The definitions given above can be instantiated by giving an information state frame to be temporalized. For instance, if we take IS^{ep} then we get a branching time variant of TEL. The C operator can then be replaced by the K operator: a temporal model $\mathcal{M}$ is then a mapping $T \to \mathsf{IS}^{ep}$, and for a propositional formula α and time point $t \in \mathbb{N}$ we have: $(\mathcal{M}, t) \models C\alpha \Leftrightarrow \alpha \in \mathrm{Th}(\mathcal{M}_t) \Leftrightarrow \mathcal{M}_t \models_{\mathrm{S5}} K\alpha$. If we take the information state frame of three-valued models, then the C-operator here has the same meaning as the C-operator of Section 4.2.

In the rest of this section, we will explore some connections between linear time and branching time semantics.

4.3.2 Homomorphisms and persistency

As mentioned before, we assume the models to be forests satisfying successor existence. In this section $\mathcal{M}$ and $\mathcal{M}'$ denote temporal models based on the flows of time $(T, <)$ and $(T', <')$ respectively. As we are interested in linear and branching time models, we need a way of relating models and we will do this using a special class of functions between models, called homomorphisms. Let two information states $m, n \in \mathsf{IS}$ be *equivalent*, denoted by $m \equiv n$, if $\mathrm{Th}(m) = \mathrm{Th}(n)$.

Definition 4.23 (Homomorphism) A (total) function $f : T \to T'$ is called a *homomorphism* of $\mathcal{M}$ to $\mathcal{M}'$ if

1. $s < t \Rightarrow f(s) < f(t)$;
2. $\mathcal{M}(s) \equiv \mathcal{M}'(f(s))$;
3. If s is a minimal element of T then $f(s)$ is minimal element of T'.

A homomorphism preserves the temporal ordering $<$, information states (up to equivalence), and minimal elements. Intuitively, a homomorphism can embed a model in a bigger model and it can identify points which have the same (up to isomorphism) path from their corresponding minimal elements. If a branching occurs at a certain point in time and there are equivalent information states at a number of next points, then we can defer the branching at this point by identifying these next points. If such a situation does not occur in a model (we shall later call such a model closed) then a homomorphism with this model as its domain can only be injective (in the branching time logic CTL a structure with this property is called *deterministic*, see [GK94]).

Lemma 4.24 Let $f : \mathcal{M} \to \mathcal{M}'$ be a homomorphism.

1. The following conditions are satisfied:
   (a) For all $t \in T$ and $s' \in T'$ with $s' < f(t)$ there exists an $s \in T$ with $s < t$ and $f(s) = s'$.
   (b) For all $t \in T$ and $s' \in T'$ with $s' \ll f(t)$ there exists an $s \in T$ with $s \ll t$ and $s' = f(s)$.
   (c) For all $s, t \in T$ it holds:
   $f(s) < f(t)$ iff there exists a $u < t$ with $f(u) = f(s)$.

   (d) For all $s, t \in T$ it holds:
   $f(s) \ll f(t)$ iff there exists a $u \ll t$ with $f(u) = f(s)$.
   (e) If $s' \in T'$ is not in the image of f, then all t' with $s' \ll t'$ are not in the image either.

2. The following are equivalent:
   (a) f is injective.
   (b) for all $s, t \in T$ it holds that $s < t$ if and only if $f(s) < f(t)$.
3. Let $t \in T$ be given with path P from a minimal element r to t. Then $f[P]$ is the path from $f(r)$ to $f(t)$ and f is a bijection between P and $f[P]$.
4. f is a surjective homomorphism to the submodel $f[\mathcal{M}] \equiv \mathcal{M}'|_{f[T]}$ of $\mathcal{M}'$.
5. If B is a branch in $\mathcal{M}$ then f is injective on B, and $f[B]$ is a branch of $\mathcal{M}'$; the restriction $f|_B$ of f to B is an isomorphism from B onto $f[B]$.

Proof: We only prove some of the above statements.
1a. Suppose $s' < f(t)$, then $f(t)$ is not minimal, therefore t is not minimal in $\mathcal{M}$ and thus has a (unique) predecessor s and therefore $f(s) < f(t)$ and $f(s) = s'$.
1e. Suppose $s' < t'$ and $t' = f(t)$. From 1a it follows that $f(s) = s'$ for some s. Therefore all immediate successors of s' are not in the image, and by induction none of the t' with $s' \ll t'$ are in the image of f. The other parts of the proof are similar.
2. For any homomorphism f it holds that $s < t \Rightarrow f(s) < f(t)$ for all s and t, so suppose also $f(s) < f(t) \Rightarrow s < t$ for all s and t, but f not injective. Then there exist $s, t \in T$ with $f(s) = f(t)$, which can be taken at minimal depth (distance from the minimal elements); this depth is the same for s as it is for t, equal to the depth of $f(s)$. If s and t are root of their components, then there are s', t' with $s < s'$ and $t < t'$, and thus $f(s) < f(s')$ and $f(t) < f(t')$, but as $f(s) = f(t)$ we also have $f(s) < f(t')$ from which it follows that $s < t'$ which is impossible since they are in different components. Let s and t now not be minimal elements. Then there are s', t' with $s' < s$ and $t' < t$ but $f(s') \neq f(t')$, since s and t were at minimal depth. But then $f(s')$ and $f(t')$ are both predecessors of $f(s)$, which is impossible in a tree. Now suppose f is injective and suppose we have s, t with $f(s) < f(t)$. Then t is not a root, so it has a predecessor t', and then $f(t') < f(t)$ so it must hold that $f(s) = f(t')$ but then $s = t'$ and therefore $s < t$. □

We are interested in preservation of truth of formulae under these homomorphisms:

Definition 4.25 Let $f : \mathcal{M} \to \mathcal{M}'$ be a homomorphism.

1. The *forward persistency* property for a formula α (under f) is defined by

$$(\mathcal{M}, t) \models \alpha \Rightarrow (\mathcal{M}', f(t)) \models \alpha$$

	H	P	∃F	∀F	∃G	∀G	∃X	∀X
preserves forward persistency	+	+	+	−	+	−	+	−
preserves backward persistency	+	+	−	+	−	+	−	+

Table 4.1: Preservation of persistency.

for all time points $t \in T$.
The *backward persistency* property for a formula α (under f) is defined by

$$(\mathcal{M}, t) \models \alpha \Leftarrow (\mathcal{M}', f(t)) \models \alpha$$

for all time points $t \in T$.
If α is both forward and backward persistent, we call it *two-sided persistent.*

2. We say a logical connective X or temporal operator Y *preserves* forward (backward) persistency (under f) if for any forward (backward) persistent formula(e) α and β (under f) also the formulae $\alpha X \beta$, $X\alpha$, $Y(\alpha)$ are forward (backward) persistent (under f).
We say a logical connective X or temporal operator Y *reverses* forward (backward) persistency (under f) if for any forward (backward) persistent formula(s) α and β (under f) the formulae $\alpha X \beta$, $X\alpha$, $Y(\alpha)$ are backward (forward) persistent (under f).

The following theorem gives an overview of all preservation properties with respect to persistent formulae (see also Table 4.1).

Theorem 4.26 Let $f : \mathcal{M} \rightarrow \mathcal{M}'$ be a homomorphism.

1. Any temporal atom $C\alpha$ is two-sided persistent under f.

2. The temporal operators H and P preserve forward and backward persistency under f.

3. The temporal operators ∃F, ∃G and ∃X preserve forward persistency, but not backward persistency under f. The temporal operators ∀F, ∀G and ∀X preserve backward persistency, but not forward persistency under f.

4. The logical connectives $\wedge$ and $\vee$ on temporal formulae preserve both forward and backward persistency under f. The logical connective $\neg$ on a temporal formula reverses forward and backward persistency under f.
If the temporal formula α is backward (forward) persistent and β forward (backward) persistent then $\alpha \rightarrow \beta$ is forward (backward) persistent (under f).

Proof:

1. This is trivial, since $\mathcal{M}'(f(t)) \equiv \mathcal{M}(t)$ for all $t \in T$.

2. For the operator P we do the following. Suppose α is forward persistent and $(\mathcal{M}, t) \models P(\alpha)$. Then for some s with $s \ll t$ it holds $(\mathcal{M}, s) \models \alpha$. By forward persistency of α we have $(\mathcal{M}', f(s)) \models \alpha$. From $s \ll t$ it follows $f(s) \ll f(t)$. So there exists an $s' \ll f(t)$ with $(\mathcal{M}', s') \models \alpha$, i.e., $(\mathcal{M}', f(t)) \models P(\alpha)$.
   Next the case of α backward persistent: from $(\mathcal{M}', f(t)) \models P(\alpha)$ it follows that there exists an $s' \in T'$ with $s' \ll f(t)$ such that $(\mathcal{M}', s') \models \alpha$.
   From Lemma 4.24 it follows that there is an $s \in T$ with $s' = f(s)$ and $s \ll t$. Now we can apply the backward persistency of α and conclude that $(\mathcal{M}, s) \models \alpha$ and so $(\mathcal{M}, t) \models P(\alpha)$. The proof for H is similar.

3. Suppose α is forward persistent and $(\mathcal{M}, s) \models \exists\mathrm{F}(\alpha)$. Then there is some $t \in T$ with $s \ll t$ such that $(\mathcal{M}, t) \models \alpha$. This implies $f(s) \ll f(t)$ and $(\mathcal{M}', f(t)) \models \alpha$ and therefore $(\mathcal{M}', f(s)) \models \exists\mathrm{F}(\alpha)$.
   - Suppose α is forward persistent and $(\mathcal{M}, s) \models \exists\mathrm{G}(\alpha)$. So there is a branch B in $\mathcal{M}$ with s on B and for all $t \in B$ with $s \ll t$ it holds $(\mathcal{M}, t) \models \alpha$. Then $B' := f[B]$ is a branch with $f(s) \in B'$. Now take a point $t' \in B$' with $f(s) \ll t'$, say $t' = f(t)$, then $s \ll t$ and therefore $(\mathcal{M}, t) \models \alpha$. The forward persistency of α ensures that $(\mathcal{M}', f(t)) \models \alpha$, so $(\mathcal{M}', t') \models \alpha$. It follows that $(\mathcal{M}', f(s)) \models \exists\mathrm{G}(\alpha)$.
   The operators $\forall\mathrm{G}$ and $\forall\mathrm{F}$ work similar (but reversed). Also the proofs for the operators $\exists\mathrm{X}$ and $\forall\mathrm{X}$ are similar.
   The homomorphism in Figure 4.1 shows the negative results.

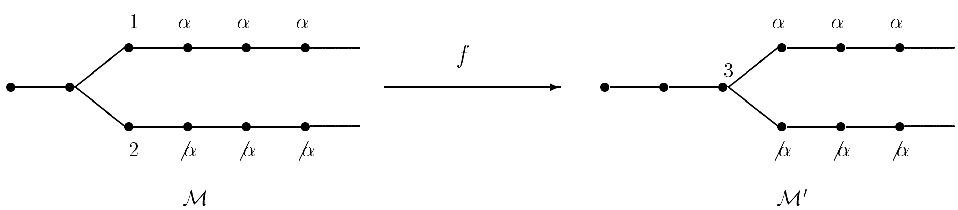

Figure 4.1: Example for negative transfer results.

   In this picture, $\mathcal{M}_1 \equiv \mathcal{M}_2$, $f(1) = f(2) = 3$ and α is a formula of $\mathcal{L}_0$ true in the upper models, but not true in the lower ones. Now $C\alpha$ is two-sided persistent, and $(\mathcal{M}, 1) \models \forall\mathrm{F}(C\alpha)$, $\forall\mathrm{G}(C\alpha)$ and $\forall\mathrm{X}(C\alpha)$, but $(\mathcal{M}', f(1)) \not\models \forall\mathrm{F}(C\alpha)$, $\forall\mathrm{G}(C\alpha)$ and $\forall\mathrm{X}(C\alpha)$, so these formulae are not forward persistent. Also, $(\mathcal{M}', f(2)) \models \exists\mathrm{F}(C\alpha)$, $\exists\mathrm{G}(C\alpha)$ and $\exists\mathrm{X}(C\alpha)$ but $(\mathcal{M}, 2) \not\models \exists\mathrm{F}(C\alpha)$, $\exists\mathrm{G}(C\alpha)$ and $\exists\mathrm{X}(C\alpha)$, so these formulae are not backward persistent.

4. We show how the connective $\neg$ works. Suppose the temporal formula α is backward persistent, and assume $(\mathcal{M}, t) \models \neg\alpha$, then $(\mathcal{M}, t) \not\models \alpha$ and because

α is backward persistent we have $(\mathcal{M}', f(t)) \not\models \alpha$ whence $(\mathcal{M}', f(t)) \models \neg\alpha$. So $\neg\alpha$ is forward persistent. The proof for the other case is analogous.

□

Theorem 4.26 can be used to build up formulae that are forward or backward persistent. For instance, for a formula $\varphi \in \mathcal{L}_0$, the formula $\exists\mathrm{F}(\neg P(\forall\mathrm{G}(\neg\exists\mathrm{F}(C\varphi))))$ is forward persistent, whereas $\exists\mathrm{F}(P(\neg\forall\mathrm{G}(\exists\mathrm{F}(C\varphi))))$ in general is not. Another example: the formula $C\varphi \rightarrow \forall\mathrm{G}(C\varphi)$, expressing conservativity, is not forward persistent. The homomorphism in Figure 4.2 shows this. However, conservativity can be defined by the set of persistent formulae $P(C\varphi) \rightarrow C\varphi$ for all $\varphi \in \mathcal{L}_0$.

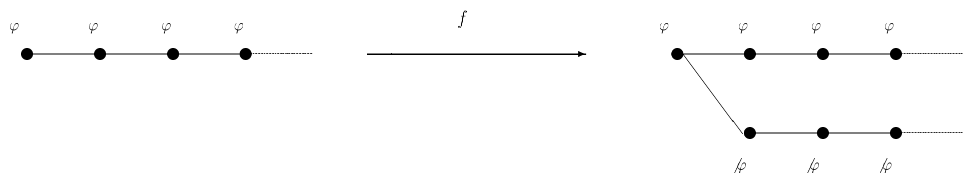

Figure 4.2: Example: conservativity does not transfer.

Theorem 4.27 Let $f : \mathcal{M} \rightarrow \mathcal{M}'$ be a homomorphism.
If α is backward persistent then

$$\mathcal{M}' \models \alpha \Rightarrow \mathcal{M} \models \alpha.$$

If f is surjective and α is forward persistent, then

$$\mathcal{M} \models \alpha \Rightarrow \mathcal{M}' \models \alpha.$$

If f is surjective and α is two-sided persistent then

$$\mathcal{M} \models \alpha \Leftrightarrow \mathcal{M}' \models \alpha.$$

So our notion of homomorphism (as we will see, strong enough to perform the algebraic constructions we have in mind) is too weak to ensure preservation of truth for all formulae. As the example in the proof of Theorem 4.26 shows, requiring only surjectivity is not enough. When identifying two points, there may be more branches through the image than through either of the two points, destroying truth of some formulae. So we need a stronger requirement:

Definition 4.28 A homomorphism $f : \mathcal{M} \rightarrow \mathcal{M}'$ is called *branch-surjective* if for all $t \in T$ and $B' \in \mathsf{Br}(T')$: if $f(t) \in B'$ then there exists a branch $B \in \mathsf{Br}(\mathcal{M})$ such that

$t \in B$ and $f[B] = B'$. A homomorphism which is surjective and branch-surjective is called *strongly branch-surjective.*

As the definition suggests, branch-surjectivity does not imply surjectivity. If $\mathcal{M}'$ consists of only one component, then this *is* the case. Branch-surjective homomorphisms preserve truth:

Proposition 4.29 For a branch-surjective homomorphism $f : \mathcal{M} \to \mathcal{M}'$, a temporal formula φ and $t \in T$:

$$(\mathcal{M}, t) \models \varphi \Leftrightarrow (\mathcal{M}', f(t)) \models \varphi.$$

Proof: It is easy to show that the operators ∃F, ∃G and ∃X preserve backward persistency under branch-surjective homomorphisms and that the operators ∀F, ∀G and ∀X preserve forward persistency under branch-surjective homomorphisms. Since then any operator preserves two-sided persistency under branch-surjective homomorphisms, all formulae must be two-sided persistent under branch-surjective homomorphisms. □

In the literature there are some different notions of homomorphism. In [Ben91b], a homomorphism is a surjective function which preserves $<$ (defined on flows of time, not on models). Thus a surjective homomorphism in our sense corresponds to a homomorphism which maps minimal elements to minimal elements in Van Benthem's sense. In [Ben91b] also the notion of a p-morphism is defined as a homomorphism which satisfies the additional "backward clause":

$$\forall t_1 \in T, t' \in T'(f(t_1) < t' \Rightarrow \exists t_2 \in T(t_1 < t_2 \land f(t_2) = t'));$$

$$\forall t_1 \in T, t' \in T'(t' < f(t_1) \Rightarrow \exists t_2 \in T(t_2 < t_1 \land f(t_2) = t')).$$

The second part of this clause is satisfied by our homomorphisms (see Lemma 4.24, 1a), and implies that minimal elements are mapped to minimal elements. The first part is equivalent to branch-surjectivity. So our notion of branch-surjective homomorphism is equivalent to the notion of p-morphism (between forests) in [Ben91b].

Similar notions (between structures) can also be defined for CTL* (see for instance [GK94]). Loosely speaking, a homomorphism from $\mathcal{M}$ to $\mathcal{M}'$ in our sense corresponds to a simulation relation from $\mathcal{M}$ to $\mathcal{M}'$ ([GK94]). They have a similar result as Theorem 4.27 for the CTL* fragment containing only ∀, respectively ∃.

We intend to use homomorphisms in a number of algebraic constructions on models, combining linear models into branching time models, and combining branching time models.

4.3.3 Algebraic constructions on temporal models

In this section, we will describe a number of constructions on temporal models. The names for these constructions are taken from [ET94a], where they have a general category-theoretic definition. There are two basic constructions. The first one aggregates a number of models into one branching time model, which is called the *coproduct* of those models.

Definition 4.30 (Coproduct) Let $(\mathcal{M}_i)_{i \in I}$ be a set of branching time models. Then the *coproduct* $\mathcal{P}$ of this set is a branching time temporal model such that

1. The flow of time of $\mathcal{P}$ is the disjoint union of the flows of time of $(\mathcal{M}_i)_{i \in I}$.
2. For a time point t in $\mathcal{P}$, we have that $\mathcal{P}(t) = \mathcal{M}_{i_0}(t)$, where $\mathcal{M}_{i_0}$ is the unique model which contains t.

Since a branching time model is a disjoint union of trees, the disjoint union of a number of branching time models is again a branching time model (see Figure 4.3). It is easy to see that for every model $\mathcal{M}_i$, there is an injective homomorphism $\mathcal{M}_i \to \mathcal{P}$ mapping $\mathcal{M}_i$ into its copy in $\mathcal{P}$.

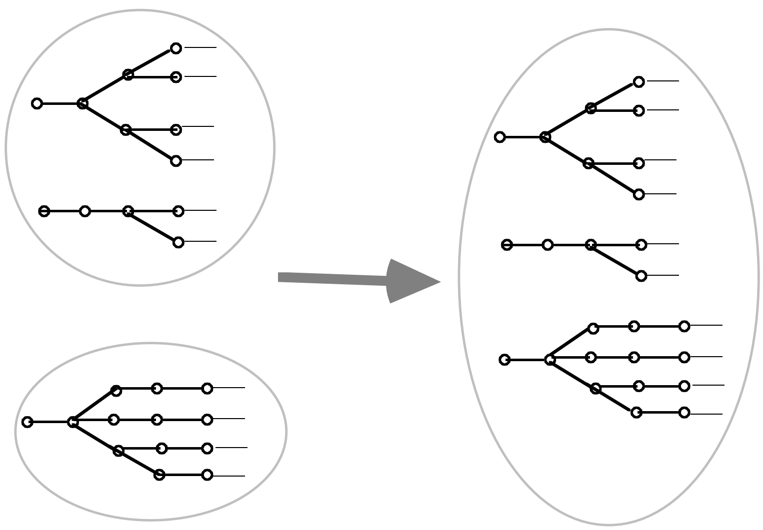

Figure 4.3: Coproduct.

The coproduct formalizes the behavior of a reasoning agent which can take either of the constituting models, but in a very inefficient way: the same branch may be in multiple models. We are interested in models which are efficient representations of behavior. Homomorphisms identify parts of a model, so a model is maximally efficient if no homomorphism can map it to a more efficient model.

Definition 4.31 (Closed model) A branching time model $\mathcal{M}$ is called *closed* if any homomorphism $f : \mathcal{M} \to \mathcal{M}'$ is injective.

There is a more direct criterion for closedness.

Proposition 4.32 Let $\mathcal{M}$ be a model, then the following are equivalent:

1. $\mathcal{M}$ is closed.
2. For all s, t and t' with $s < t$, $s < t'$ and $\mathcal{M}(t) \equiv \mathcal{M}(t')$ it holds that $t = t'$, and for minimal elements r, r' with $\mathcal{M}(r) \equiv \mathcal{M}(r')$ it holds that $r = r'$.

Proof: Suppose $\mathcal{M}$ is closed but there are $s, t \neq t'$ with $s < t$, $s < t'$ and $\mathcal{M}(t) \equiv \mathcal{M}(t')$. Define a homomorphism f on T which is identity except that $f(t) = f(t')$. Let the successor relation on $T' = f[T]$ be defined by $u < v$ iff there are u' and v' in T with $u = f(u')$ and $v = f(v')$ and $u' < v'$. Let $\mathcal{M}'$ be defined by $\mathcal{M}'(f(s)) \equiv \mathcal{M}(s)$. Since f is surjective and identity except on t and t' where $\mathcal{M}(t) = \mathcal{M}(t')$, this is well-defined. It is easy to check that the model $\mathcal{M}'$ based on $(T', <)$ is a forest. Now f is not injective, so $\mathcal{M}$ is not closed, which it was supposed to be. If there are roots r, r' with $\mathcal{M}(r) \equiv \mathcal{M}(r')$ then this same construction can be applied. Conversely, suppose the second condition is satisfied, but $\mathcal{M}$ is not closed. Then there is a homomorphism $f : \mathcal{M} \to \mathcal{M}'$ which is not injective. Therefore, there must be $t \neq t'$ with $f(t) = f(t')$. One can take these points at a minimal distance from their roots (and that is the same distance, equal to the distance to the root of $f(t) = f(t')$). If they are roots, then $\mathcal{M}(t) \equiv \mathcal{M}'(f(t)) \equiv \mathcal{M}'(f(t')) \equiv \mathcal{M}(t')$ and then the second condition is violated. If they are not roots, then still $\mathcal{M}(t) \equiv \mathcal{M}(t')$. Furthermore they have immediate predecessors $s < t$ and $s' < t'$. Then $f(s) < f(t)$ and $f(s') < f(t') = f(t)$. Then it must hold that $f(s) = f(s')$, and as above then also $\mathcal{M}(s) \equiv \mathcal{M}(s')$. But as t and t' were chosen at minimal depth it must hold that $s = s'$, and then the second condition is violated again. Therefore $\mathcal{M}$ must be closed. □

So in a closed model there are no two different minimal elements with equivalent information state, and any two different successors of a point have a non-equivalent information state.

For every model there exists a closed model into which it can be mapped by a surjective homomorphism. This model is unique up to isomorphism.

Proposition 4.33 For every branching time model $\mathcal{M}$ there exists a closed model $\mathrm{cl}(\mathcal{M})$, called the *closure of* $\mathcal{M}$, such that there is a surjective homomorphism from $\mathcal{M}$ to $\mathrm{cl}(\mathcal{M})$.

Proof: The idea in the construction of this closure is to identify common initial sub-branches (up to information state equivalence) with each other as much as possible (see Figure 4.4).

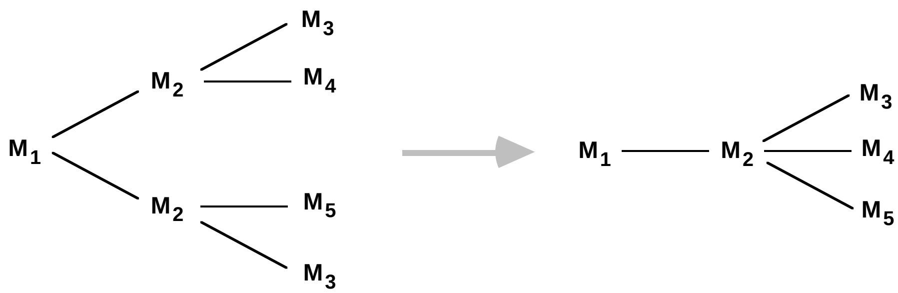

Figure 4.4: Closure.

Let $\mathcal{M}$ be based on $(T, <)$. Define an equivalence relation $\sim$ on T as follows: $s \sim t$ iff $s' < s$ and $t' < t$ and $s' \sim t'$ and $\mathcal{M}(s) \equiv \mathcal{M}(t)$, or s and t are root and $\mathcal{M}(s) \equiv \mathcal{M}(t)$. (This is an inductive definition on the depth of the points s and t.) Now let S be the set of equivalence classes $T/\sim$, define $\mathcal{N}$ on S by $\mathcal{N}([s]) \equiv \mathcal{M}(s)$ (this is well- defined), where $[s]$ is the equivalence class of s. For two equivalence classes k_1 and k_2, define $k_1 < k_2$ iff there is an s in k_1 and a t in k_2 with $s < t$. Notice that two elements can only be equivalent if they are at the same distance from the root. It is easy to check that the ordering $<$ on S is irreflexive, antisymmetric and intransitive. The proofs of well-foundedness and left-linearity of $(\mathcal{N}, S, <)$ are straightforward and omitted.

Now we will check that the mapping h which takes an element s of T to its class in S is a homomorphism:

1. if $s < t$ in T then $[s] < [t]$ by definition.
2. $\mathcal{N}([s]) \equiv \mathcal{M}(s)$ by definition.
3. Take an $s \in T$ which is a root, and suppose $[s]$ is not a root, then there is a $t \sim s$ which is not a root, but then it can not hold that $s \sim t$.

By definition h is surjective. Using Proposition 4.32, the proof of closedness of $\mathcal{N}$ is straightforward. □

Using the constructions of coproduct and closure, we can define a new construction, the joint closure. The idea is that we want to merge a number of temporal models into a new one, which is a most efficient representation of the possibilities given by these models.

Definition 4.34 Let S be a set of models. The *joint closure* jcl(S) of S is the closure of the coproduct of S.

4.3.4 Logical connections

In this subsection we investigate the logical properties of the constructions of the previous subsection. First we consider some persistency results of formulae under coproducts and joint closures. Next we will discuss four specific classes of models and introduce associated semantic satisfaction relations, and their mutual connections. One of these classes is the class of linear time models. We will discuss the connections with that class in some more detail.

Proposition 4.35 Suppose T is a disjoint union of isolated T_i, and $\mathcal{M}$ a temporal model based on T. Then for $t \in T_i$:

$$\begin{array}{lcl} (\mathcal{M}, t) \models \varphi & \Leftrightarrow & (\mathcal{M}|_{T_i}, t) \models \varphi \\ \mathcal{M} \models \varphi & \Leftrightarrow & \{\mathcal{M}|_{T_i} \mid i \in I\} \models \varphi. \end{array}$$

Proof: Straightforward. □

As an immediate consequence we have persistency under the coproduct construction.

Corollary 4.36 Let $(\mathcal{M}_i)_{i \in I}$ be a set of temporal models and P their coproduct.

1. For any formula φ and $t \in T_i$ it holds

$$\begin{array}{lcl} (P, t) \models \varphi & \Leftrightarrow & (\mathcal{M}_i, t) \models \varphi \\ P \models \varphi & \Leftrightarrow & \text{for all } i \in I \text{ it holds } \mathcal{M}_i \models \varphi. \end{array}$$

2. Let $\mathfrak{T}$ be a temporal theory, then P is a model of $\mathfrak{T}$ if and only if for every $i \in I$ the model $\mathcal{M}_i$ is a model of $\mathfrak{T}$.

Because the joint closure is built from a coproduct followed by a surjective homomorphism, and the coproduct construction behaves well under persistency, we have persistency under joint closure in the following sense.

Corollary 4.37 Let C be the joint closure of the indexed set of models $(\mathcal{M}_i)_{i \in I}$, where $\mathcal{M}_i$ is based on T_i. For $t \in T_i$, t' denotes the corresponding time point in the joint closure.
Every formula φ that is forward persistent under surjective homomorphisms satisfies

$$\begin{array}{lcl} (\mathcal{M}_i, t) \models \varphi & \Rightarrow & (C, t') \models \varphi \\ \mathcal{M}_i \models \varphi \text{ for all } i \in I & \Rightarrow & C \models \varphi. \end{array}$$

Every formula φ that is backward persistent under surjective homomorphisms satisfies

$$\begin{array}{lcl} (C, t') \models \varphi & \Rightarrow & (\mathcal{M}_i, t) \models \varphi \\ C \models \varphi & \Rightarrow & \mathcal{M}_i \models \varphi \text{ for all } i \in I. \end{array}$$

Every formula φ that is two-sided persistent under surjective homomorphisms satisfies

$$\begin{array}{lcl} (C,t') \models \varphi & \Leftrightarrow & (\mathcal{M}_i,t) \models \varphi \\ C \models \varphi & \Leftrightarrow & \mathcal{M}_i \models \varphi \text{ for all } i \in I. \end{array}$$

In the class BT of all branching time models we distinguish two subclasses, namely LT, the class of linear time models and CL, the class of closed models. Since it is easy to establish that linear time models are closed we have

$$\mathsf{LT} \subset \mathsf{CL} \subset \mathsf{BT}.$$

There are other connections as well. Every branching time model can be mapped by a surjective homomorphism onto a closed one. Moreover, all branches in a branching time model are linear models, and together they cover the whole flow of time. For any set of models S, its joint closure $\mathrm{jcl}(S)$ can be constructed. From these classes of models we can define corresponding satisfaction relations.

Definition 4.38 Let S be a class of branching time temporal models. For any temporal formula φ, define:

$$\begin{array}{lcl} S \models_{\mathsf{BT}} \varphi & \Leftrightarrow & (\forall \mathcal{M} \in \mathsf{BT} : \mathcal{M} \in S \Rightarrow \mathcal{M} \models \varphi) \\ S \models_{\mathsf{CL}} \varphi & \Leftrightarrow & (\forall \mathcal{M} \in \mathsf{CL} : \mathcal{M} \in S \Rightarrow \mathcal{M} \models \varphi) \\ S \models_{\mathsf{LT}} \varphi & \Leftrightarrow & (\forall \mathcal{M} \in \mathsf{LT} : \mathcal{M} \in S \Rightarrow \mathcal{M} \models \varphi) \\ S \models_{\mathsf{JCL}} \varphi & \Leftrightarrow & \mathrm{jcl}(S) \models \varphi. \end{array}$$

Obviously, we have that, for instance, $S \models_{\mathsf{LT}} \varphi \Leftrightarrow S \cap \mathsf{LT} \models_{\mathsf{BT}} \varphi$. These definitions may seem rather unusual, but are probably more familiar when considering the interesting case when S is the set of all branching time models of a temporal theory $\mathfrak{T}$. Then $S \models_{\mathsf{BT}} \varphi$ means that φ is a branching time semantical consequence of $\mathfrak{T}$. Furthermore, $S \models_{\mathsf{LT}} \varphi$ means that φ is a linear time consequence of $\mathfrak{T}$.

There are some apparent logical relations between these notions:

Proposition 4.39 Let S be a class of branching time temporal models. Then the following holds for all temporal formulae φ:

$$S \models_{\mathsf{BT}} \varphi \Rightarrow S \models_{\mathsf{CL}} \varphi \Rightarrow \left\{ \begin{array}{l} S \models_{\mathsf{LT}} \varphi \\ S \models_{\mathsf{JCL}} \varphi. \end{array} \right.$$

Proof: This is an easy consequence of the definition of the notions $\models_{\mathsf{BT}}$, $\models_{\mathsf{CL}}$, $\models_{\mathsf{LT}}$, and $\models_{\mathsf{JCL}}$, together with the former observation that $\mathsf{LT} \subset \mathsf{CL} \subset \mathsf{BT}$ and the fact that $\mathrm{jcl}(S) \in \mathsf{CL}$ for any S (which follows from its definition, Definition 4.34, and the definition of closure in Proposition 4.33). $\square$

A main question is how different these four notions are, and, in general, what the relations are. There is a real difference between linear time models and the others because they satisfy the following axioms expressing indistinguishable future:

$$\begin{aligned} \exists X\varphi &\leftrightarrow \forall X\varphi \\ \exists F\varphi &\leftrightarrow \forall F\varphi \\ \exists G\varphi &\leftrightarrow \forall G\varphi. \end{aligned}$$

The joint closure of a set of (linear) time models will not in general satisfy these axioms. Any branching time model satisfying these axioms can in fact be mapped uniquely to a linear one.

First we need the following connection between a model $\mathcal{M}$ and its collection of linear time submodels (its branches) $\mathsf{Br}(\mathcal{M})$. This notion is extended to a class of models S: the set of all branches of models in S is denoted by $\mathsf{Br}(S)$. In the following theorem isomorphism ($\equiv$) is a relation among temporal models. It means that the flows of time are isomorphic with corresponding (equivalent) information states.

Theorem 4.40 Let $\mathcal{M}$ be any model. Then $\mathrm{jcl}(\mathsf{Br}(\mathcal{M})) \equiv \mathrm{cl}(\mathcal{M})$. In particular, $\mathcal{M}$ is closed if and only if $\mathrm{jcl}(\mathsf{Br}(\mathcal{M})) \equiv \mathcal{M}$.

Proof: See [ET94a]. □

A model $\mathcal{M}$ in a set of models S is called a *final model* in S, if for every model $\mathcal{M}' \in S$ there is a unique homomorphism $f : \mathcal{M}' \to \mathcal{M}$. We have the following result on the existence of final models of a theory. For a class of models S the class of models S^* is defined by

$$S^* = \{\mathrm{jcl}(S') \mid S' \subseteq S\}.$$

In particular, S^* contains the joint closure $\mathrm{jcl}(S)$ of all models in S, but it also contains the closure of each individual model: for $\mathcal{M} \in S$, we have that $\mathrm{jcl}(\{\mathcal{M}\}) = \mathrm{cl}(\mathcal{M})$.

Theorem 4.41 Let $\mathfrak{T}$ be a temporal theory that is forward persistent under surjections and S a set of models of $\mathfrak{T}$. Then S^* is a set of models of $\mathfrak{T}$ and the joint closure $\mathrm{jcl}(S)$ of all models of S is a final model of $\mathfrak{T}$ in S^*.

Proof: See [ET94a]. □

A class of models S is *closed under submodels* if for each model in the class, all of its submodels are also in the class. In particular, in that case we have $\mathsf{Br}(S) \subseteq S$. A class S is *closed under surjections* if whenever $\mathcal{M}$ is in S and $f : \mathcal{M} \to \mathcal{N}$ is a surjective homomorphism, $\mathcal{N}$ is also in S. After these preparations we are able to establish the following theorem that gives more precise connections between the different semantic satisfaction relations.

Theorem 4.42 Let S be a class of models, and φ any formula.

1. If S is closed under submodels, and φ is forward persistent under surjections, then

$$S \models_{\mathsf{BT}} \varphi \Leftrightarrow S \models_{\mathsf{LT}} \varphi.$$

2. If S is closed under surjections and φ is backward persistent under surjections, then

$$S \models_{\mathsf{BT}} \varphi \Leftrightarrow S \models_{\mathsf{CL}} \varphi.$$

3. If S is a set, then

$$S^* \models_{\mathsf{BT}} \varphi \Rightarrow S \models_{\mathsf{JCL}} \varphi.$$

   If, moreover, φ is backward persistent, then

$$S^* \models_{\mathsf{BT}} \varphi \Leftrightarrow S \models_{\mathsf{JCL}} \varphi.$$

4. If φ is forward persistent, then

$$\mathsf{Br}(S) \models_{\mathsf{LT}} \varphi \Rightarrow S \models_{\mathsf{BT}} \varphi.$$

   If, moreover, $\mathsf{Br}(S) \subseteq S$, then

$$S \models_{\mathsf{LT}} \varphi \Leftrightarrow \mathsf{Br}(S) \models_{\mathsf{LT}} \varphi \Leftrightarrow S \models_{\mathsf{BT}} \varphi.$$

   If, in addition, $\mathsf{Br}(S^*) = \mathsf{Br}(S)$, then

$$S \models_{\mathsf{LT}} \varphi \Leftrightarrow S \models_{\mathsf{BT}} \varphi \Leftrightarrow S^* \models_{\mathsf{BT}} \varphi.$$

5. If S is a set and $\mathsf{Br}(S) = \mathsf{Br}(S^*) \subseteq S$ and φ is both forward and backward persistent, then

$$S \models_{\mathsf{LT}} \varphi \Leftrightarrow S \models_{\mathsf{JCL}} \varphi.$$

Proof:

1. Assume $S \models_{\mathsf{LT}} \varphi$. Suppose an $\mathcal{M}$ in S is given. Because S is closed under submodels, $\mathsf{Br}(\mathcal{M}) \models \varphi$. By forward persistency of φ we also have $\mathcal{M} \models \varphi$. We have proven

$$S \models_{\mathsf{BT}} \varphi \Leftrightarrow S \models_{\mathsf{LT}} \varphi.$$

2. Assume $S \models_{\mathsf{CL}} \varphi$. Let $\mathcal{M}$ in BT be given with $\mathcal{M}$ in S. Then there is a surjective homomorphism of $\mathcal{M}$ onto its closure $\mathrm{cl}(\mathcal{M})$ in CL. Because S is closed under surjective homomorphisms we have $\mathrm{cl}(\mathcal{M}) \in S$, and therefore $\mathrm{cl}(\mathcal{M}) \models \varphi$. By persistency of φ we have $\mathcal{M} \models \varphi$. We have proven

$$S \models_{\mathsf{BT}} \varphi \Leftrightarrow S \models_{\mathsf{CL}} \varphi.$$

3. The first implication is trivial. Assume φ is backward persistent and $S \models_{\mathsf{JCL}} \varphi$. Let $\mathcal{M}$ be any model in S^*. Then we can map $\mathcal{M}$ in $\mathrm{jcl}(S)$. Since $\mathrm{jcl}(S) \models \varphi$, and φ is backward persistent we have $\mathcal{M} \models \varphi$. Therefore

$$S \models_{\mathsf{BT}} \varphi \Leftrightarrow S \models_{\mathsf{JCL}} \varphi.$$

4. Suppose φ is forward persistent, and $\mathsf{Br}(S) \models_{\mathsf{LT}} \varphi$ then every branch in every model in S satisfies φ, and can be mapped into this model. Therefore by persistence $S \models_{\mathsf{BT}} \varphi$.

   Assume, moreover, $\mathsf{Br}(S) \subseteq S$, then it is trivial that $S \models_{\mathsf{BT}} \varphi$ implies $\mathsf{Br}(S) \models_{\mathsf{LT}} \varphi$, and that this is equivalent to $S \models_{\mathsf{LT}} \varphi$.

   Assume, in addition, $\mathsf{Br}(S^*) = \mathsf{Br}(S)$, then the previous result can be applied to S^*. It follows that

$$S^* \models_{\mathsf{LT}} \varphi \Leftrightarrow \mathsf{Br}(S^*) \models_{\mathsf{BT}} \varphi \Leftrightarrow S^* \models_{\mathsf{BT}} \varphi.$$

5. This follows from the two previous points.

$\square$

In Corollary 4.37 and Theorems 4.41 and 4.42, properties are established for formulae which are forward or backward persistent under (possibly surjective or injective) homomorphisms. The question arises whether there are formulae persistent under these special homomorphisms but not under any homomorphism. This turns out to be not the case:

Proposition 4.43 A formula is forward (backward) persistent under surjective / injective homomorphisms if and only if it is forward (backward) persistent under any homomorphism.

Proof:
• First we will prove the case for forward persistency under surjections. Suppose φ is forward persistent under surjective homomorphisms, but not under any homomorphism. Then there is a (non-surjective) homomorphism $f : \mathcal{M} \to \mathcal{M}'$ with $t \in T$ such that $(\mathcal{M}, t) \models \varphi$ but $(\mathcal{M}', f(t)) \not\models \varphi$. Now construct the model $\mathcal{N}$ which consists of a copy of $\mathcal{M}$ and for every point s of $\mathcal{M}'$ not in the image of f a branch B of $\mathcal{M}'$ with $s \in B$ (disjoint from the copy of $\mathcal{M}$ and disjoint from every other

such branch). Then it is easy to see that $(\mathcal{N}, t) \models \varphi$. Let $g : \mathcal{N} \to \mathcal{M}'$ be the function which maps the copy of $\mathcal{M}$ to $f[\mathcal{M}]$ in $\mathcal{M}'$, and which maps every added branch to the same branch in $\mathcal{M}'$. Then g is a homomorphism and g is surjective and $(\mathcal{M}', g(t)) \not\models \varphi$. This is in contradiction with the assumption that φ was forward persistent under surjective homomorphisms. Therefore φ is forward persistent under any homomorphism. The proof for backward persistency under surjective homomorphisms uses the same construction.

• Now for the case of backward persistency under injective homomorphisms. Suppose φ is backward persistent under injective homomorphisms, but not under any homomorphism. Then there is a (non-injective) homomorphism $f : \mathcal{M} \to \mathcal{M}'$ with $t \in T$ such that $(\mathcal{M}', f(t)) \models \varphi$ but $(\mathcal{M}, t) \not\models \varphi$. Now construct a model $\mathcal{N}$ by taking a copy of $\mathcal{M}$ and adding the following: for every point s of $\mathcal{M}$, and for every branch B' of $\mathcal{M}'$ with $f(s) \in B'$, if there is no equivalent branch B in $\mathcal{M}$ with $s \in B$, then we add such a branch in $\mathcal{N}$. Now let $h : \mathcal{M} \to \mathcal{N}$ be the injective homomorphism which maps $\mathcal{M}$ to its copy in $\mathcal{N}$. Let $g : \mathcal{N} \to \mathcal{M}'$ be the homomorphism (!) which maps $h(s)$ to $f(s)$ and the branches from $\mathcal{M}'$ in $\mathcal{N}$ to their counterparts in $\mathcal{M}'$. Of course the model $\mathcal{N}$ and the homomorphism g are constructed in such a way that g is branch-surjective, and we can use Proposition 4.29: since $(\mathcal{M}', f(t)) \models \varphi$ and $f(t) = g(h(t))$ and g is branch-surjective, we have $(\mathcal{N}, h(t)) \models \varphi$, but as h is injective and φ is backward persistent under injections we have $(\mathcal{M}, t) \models \varphi$, contradicting the assumption. Thus φ must be backward persistent under any homomorphism. The proof for forward persistency under injective homomorphisms uses the same construction. □

Theorem 4.42 describes some cases in which the different satisfaction relations are equal. The question still remains whether they are not in general always equal. This is not the case:

Proposition 4.44 In general $\models_{\mathsf{BT}} \neq \models_{\mathsf{CL}}$, $\models_{\mathsf{CL}} \neq \models_{\mathsf{LT}}$, $\models_{\mathsf{CL}} \neq \models_{\mathsf{JCL}}$. Moreover, for each of these inequalities we can find a temporal theory $\mathfrak{T}$ such that for some φ we have $Mod(\mathfrak{T}) \models_X \varphi$ but not $Mod(\mathfrak{T}) \models_Y \varphi$.

Proof: We have already remarked that $S \models_{\mathsf{LT}} \exists \mathrm{X}\varphi \leftrightarrow \forall \mathrm{X}\varphi$ for any class S; it is easy to see that this does not hold for the other consequence relations. Let us look at $\models_{\mathsf{BT}}$ and $\models_{\mathsf{CL}}$. Take any model $m \in \mathsf{IS}$ and let $\alpha \in \mathcal{L}_0$ be such that there exist $k, l \in \mathsf{IS}$ with $\alpha \in \mathrm{Th}(k)$ and $\alpha \notin \mathrm{Th}(l)$ (thus the information state frame should not be trivial). Define the following two formulae:

$$at_0 := H(\bot)$$
$$at_1 := P(\top) \wedge HH(\bot).$$

It is easy to see that at_0 is true in a point if and only if it is a minimal element and at_1 is true in a point if and only if it is a successor of a minimal element. Now define

$$\begin{aligned}\mathfrak{T} &:= \{at_i \to C\varphi \mid \varphi \in \mathrm{Th}(m), \varphi \in \mathcal{L}_0, i = 0, 1\} \cup \\ &\quad\ \{at_i \to \neg C\varphi \mid \varphi \notin \mathrm{Th}(m), \varphi \in \mathcal{L}_0, i = 0, 1\} \cup \\ &\quad\ \{at_0 \to \exists\mathrm{F}(\exists\mathrm{F}(C\alpha))\}, \text{ and} \\ \varphi &:= at_1 \to \exists\mathrm{F}(C\alpha).\end{aligned}$$

If $\mathcal{M}$ is a closed model of $\mathfrak{T}$ then all initial points and their immediate successors must be equivalent to m, but as $\mathcal{M}$ is closed, it must have a unique root r with one immediate successor s. Then $(\mathcal{M}, r) \models \exists\mathrm{F}(\exists\mathrm{F}(C\alpha))$, so there exists a point t with $r \ll t$ and $(\mathcal{M}, t) \models \exists\mathrm{F}(C\alpha)$, but then $(\mathcal{M}, s) \models \exists\mathrm{F}(C\alpha)$ since s is the only immediate successor of r. Thus $\mathcal{M} \models \varphi$, and we have proved $Mod(\mathfrak{T}) \models_{\mathsf{CL}} \varphi$. Now consider the branching time model in Figure 4.5.

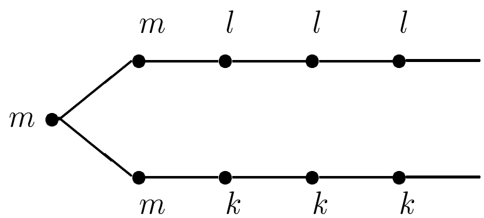

Figure 4.5: Counterexample.

In this model, the minimal element and its immediate successors are equivalent to m and a k state is reachable from the root in at least two steps (remember that $\alpha \in \mathrm{Th}(k)$), so this model is a model of $\mathfrak{T}$. But the upper successor of the root is a successor of a minimal element but has only l states reachable (in which $C\alpha$ is not true), so it is not a model of φ. We have proven that $Mod(\mathfrak{T}) \not\models_{\mathsf{BT}} \varphi$.

Now we will look at $\models_{\mathsf{CL}}$ and $\models_{\mathsf{JCL}}$. Let $m \in \mathsf{IS}$ and let $\alpha \in \mathcal{L}_0$ be such that there exist $k, l \in \mathsf{IS}$ with $\alpha \in \mathrm{Th}(k)$ and $\alpha \notin \mathrm{Th}(l)$ (thus the underlying information state frame should not be trivial). Define

$$\begin{aligned}\mathfrak{T} &:= \{at_0 \to C\varphi \mid \varphi \in \mathrm{Th}(m), \varphi \in \mathcal{L}_0\} \cup \\ &\quad\ \{at_0 \to \neg C\varphi \mid \varphi \notin \mathrm{Th}(m), \varphi \in \mathcal{L}_0\}.\end{aligned}$$

Let the information state frame be such that IS is a set (for instance propositional logic). Then we can take $\mathrm{jcl}(Mod(\mathfrak{T}))$ which has a unique root r (with information state equivalent to m), in which for each point the set of its immediate successors consists of one state for each information state (up to equivalence). This model contains a branch starting at r in which each point has k as its information state. So $\mathrm{jcl}(Mod(\mathfrak{T})) \models at_0 \to \exists\mathrm{F}(C\alpha)$ which gives $Mod(\mathfrak{T}) \models_{\mathsf{JCL}} at_0 \to \exists\mathrm{F}(C\alpha)$. Now consider the linear model $\mathcal{N}$ consisting of a root with information state equivalent to m and all the other points have information states equivalent to l. Then this is a closed model of $\mathfrak{T}$ but $\mathcal{N} \not\models at_0 \to \exists\mathrm{F}(C\alpha)$ so $Mod(\mathfrak{T}) \not\models_{\mathsf{CL}} at_0 \to \exists\mathrm{F}(C\alpha)$. □

Suppose we have a set of models S (possibly the set of models of some temporal theory) which describes some form of reasoning. The branches of models in S represent valid reasoning paths. Often, we are interested in formulae from $\mathcal{L}_0$ which become true in every reasoning path (we will call them *sceptical* conclusions), and formulae which become true in at least one reasoning path (we will call them *credulous* conclusions). Formally, we could define these conclusions as follows:

$$S \mathrel{|\!\!\approx}_{\text{scep}} \alpha \quad \Leftrightarrow \quad S \models_{\mathsf{BT}} \forall \mathrm{F}(C\alpha)$$
$$S \mathrel{|\!\!\approx}_{\text{cred}} \alpha \quad \Leftrightarrow \quad S \not\models_{\mathsf{BT}} \forall \mathrm{F}(\neg C\alpha)$$

for $\alpha \in \mathcal{L}_0$.

A formula $\forall\mathrm{F}(C\alpha)$ or $\forall\mathrm{F}(\neg C\alpha)$ preserves backward persistency (see Theorem 4.26), and we can use Theorem 4.42 to give some more connections. For example, if S is closed under surjections (this is the case, for instance, if S is the set of branching time models of a theory which consists of forward persistent formulae), then $S \mathrel{|\!\!\approx}_{\text{scep}} \alpha \Leftrightarrow S \models_{\mathsf{CL}} \forall\mathrm{F}C\alpha$.

4.3.5 Final remarks

Branching time temporal models can be used to describe the behavior of dynamic processes, such as the reasoning processes of reasoning agents. The linear models usually describe a possible reasoning pattern, and a set of such models can be used to describe possible patterns. These models may be described by a temporal theory. Another way of describing possible behavior is by a branching time process which branches at any time a pattern can continue in more than one way. These models can also be axiomatized by a temporal theory. In this subsection we have tried to identify a uniform manner in which to relate these different kinds of models. A number of operations like the coproduct, closure and joint closure which perform a kind of merging of models into a final model were defined. Therefore, out of a set of linear models we can construct a branching time model which incorporates all the linear models. This can then be transformed by homomorphisms into a model which is closed. In that model all decisions that have to be made during a process (which branch to take) are moved as far backward in time as possible. This gives the most efficient representation of the multitude of possible paths.

The results of this section will be used later on (in Section 5.2) to give a branching time semantics to default logic.

In [Spa90], a reduction from linear time logic to branching time logic is given, by translating formulae from linear time logic into formulae from branching time logic. The translation replaces the F-operator by $\forall$F, and forces "linear behavior" on subformulae (meaning that $\forall\mathrm{F}\alpha$ and $\exists\mathrm{F}\alpha$ should be equivalent). The idea of viewing a linear time model as a (simple) branching time model, and the construction of the set of linear time models $\mathsf{Br}(\mathcal{M})$ of branches of the model $\mathcal{M}$, occur in [Spa90].

Theorem 4.26 in fact gives formation rules for persistent formulae. An interesting question is whether these rules generate *all* persistent formulae. Of course, the formula $\forall\mathrm{G}C\alpha \vee \neg\forall\mathrm{G}C\alpha$ can not be formed by these rules, but is equivalent to

$C\alpha \vee \neg C\alpha$ which is formed by these rules. The question can be phrased as follows: is every forward/backward persistent formula equivalent to a formula generated by the rules for forward/backward persistent formulae of Theorem 4.26?

4.4 Minimal models and minimal entailment

In order to specify reasoning in a temporal logic, one can write down a set of temporal formulae that constrain the behavior of the reasoning process. Often, such formulae prescribe that the agent must draw a certain conclusion in certain circumstances. For example, there may be a rule stating that if the agent knows α *now*, then it should know β *from the next point in time onwards*. In general, we do not wish to specify all the conclusions the agent should *not* draw in certain circumstances (this problem is similar to the frame problem in logics of action and change, see for instance [MH69], [Hay73], [Sho88]). A possible solution to this problem is to maximize the agent's ignorance over time, or to minimize the change in the agent's information state over time, given the temporal formulae it should obey. This maximization or minimization can be formalized by introducing a preference ordering on temporal models, which favors models with less knowledge. Instead of all models of the temporal description, one can restrict oneself to the minimal models of the description with respect to such an ordering. There are several alternatives possible for the definition of such an ordering. We will treat a few of them in this section. For a broader discussion of minimization of models, see for instance [Ben89].

4.4.1 Global minimality of knowledge and MTEL

We will take the logic TELC (see Definition 4.6) as a starting point. The first ordering that we will introduce maximizes the agent's ignorance over time, given the temporal formulae it should obey. This maximization can be formalized by introducing a preference ordering on temporal models, which favors models with less knowledge. So we make the explicit assumption that "all the agent knows" is what is dictated by the temporal formulae. Apart from the temporal aspect, this is similar in spirit to the theory of epistemic states of Halpern and Moses [HM85b], introduced to formalize the notion of "only knowing φ". The global minimality criterion compares temporal models pointwise.

Definition 4.45 (Minimal models and entailment)

1. We define the ordering $\preceq^g$ on TELC-models by defining for $\mathcal{M}, \mathcal{N}$:

$$\mathcal{M} \preceq^g \mathcal{N} \Leftrightarrow \text{ for all } s \in \mathbb{N} : \mathcal{M}_s \preceq \mathcal{N}_s$$

   where $\preceq$ is the ordering on IS^{ep}. We write $\mathcal{M} \prec^g \mathcal{N}$ if $\mathcal{M} \preceq^g \mathcal{N}$ and $\mathcal{M} \neq \mathcal{N}$.

2. A TELC-model $\mathcal{M}$ is a $\preceq^g$*-minimal model of* a set of formulae T, denoted $\mathcal{M} \models_{\preceq^g} T$, if

- $\mathcal{M} \models T$, and
- for any conservative model $\mathcal{N}$, if $\mathcal{N} \models T$ and $\mathcal{N} \preceq^g \mathcal{M}$ then $\mathcal{N} = \mathcal{M}$.

3. For a set of TEL-formulae T and a formula φ, we say φ is a *$\preceq^g$-minimal consequence of* T, denoted $T \models_{\preceq^g} \varphi$, if for all $\preceq^g$-minimal models $\mathcal{M}$ of T, $\mathcal{M} \models \varphi$ holds. This preferential logic is called MTEL.

The 'g' in the notation $\preceq^g$ stands for '*g*lobal'. We can use a temporal theory T to describe the reasoning process of an agent. In general, this set of formulae will dictate that the agent knows some objective formulae at certain points in time. In order to avoid having to state (within T) that many other formulae have to remain unknown to the agent (neither known to be true nor known to be false), we want to make the assumption that "all the rest is unknown as much as possible". The above ordering and minimal consequence formalize this. Thus the minimal models of T are the intended models of T, describing the intended behavior of the agent over time. We can then use minimal consequence to infer properties of this reasoning process.

Proposition 4.46 (Closedness of minimal models) Let T be a set of TEL-formulae, and let $\mathcal{M}$ be a TELC-model. If $\mathcal{M} \models_{\preceq^g} T$, then $\mathcal{M}$ is closed (see Definition 4.8).

Proof: Suppose $\mathcal{M} \models_{\preceq^g} T$. Define the model $\mathcal{N}$ by $\mathcal{N}_i = Mod(Th(\mathcal{M}_i))$ for all $i \in \mathbb{N}$. It is easy to prove that $Mod(Th(\mathcal{M}_i)) \supseteq \mathcal{M}_i$ so $\mathcal{N} \preceq^g \mathcal{M}$. Furthermore, by induction on the complexity of formulae , it is straightforward to prove that for any $i \in \mathbb{N}$, and for any formula φ it holds that $(\mathcal{M}, i) \models \varphi \Leftrightarrow (\mathcal{N}, i) \models \varphi$. So $\mathcal{N} \models T$. As $\mathcal{M}$ is a minimal model of T, it follows that $\mathcal{M} = \mathcal{N}$, which means that $\mathcal{M}_i = Mod(Th(\mathcal{M}_i))$, so $\mathcal{M}_i$ is closed for all i. We conclude that $\mathcal{M}$ is closed. $\square$

This means that Proposition 4.9 always applies to minimal models: the limit contains exactly the knowledge which is derived at some point in the temporal model.

Note that the notion of minimal entailment strengthens the notion of conservative entailment in the sense that $T \models^c \psi$ implies $T \models_{\preceq^g} \psi$. An easy example, even without temporal operators, shows that it is a proper extension: although $Kp \not\models^c \neg Kq$ we do have $Kp \models_{\preceq^g} \neg Kq$.

The formula $K\alpha \to GK\beta$ describing the rule at the beginning of this section, now has the correct (intended) models. The only minimal model of $K\alpha \to GK\beta$ itself is the sequence $\{\mathrm{Val}(P)\}_{i \in \mathbb{N}}$, in which the agent knows nothing (besides the tautologies) over time. The only minimal model of $(K\alpha \to GK\beta) \wedge K\alpha$ is the sequence $Mod(\alpha), Mod(\alpha \wedge \beta), Mod(\alpha \wedge \beta), \ldots$.

The definition of the ordering $\preceq^g$ depended on the ordering of the information states IS^{ep}. If we take the ordering $\preceq$ on partial states of IS^{3val} (see Definition 2.3), we get an ordering on TPLC-models (see Definition 4.12), and we can define minimal consequence in a completely analogous way.

4.4.2 Global minimality with equal limits

The ordering $\preceq^g$ prefers models in which the knowledge over time is minimal. A variant of this ordering also compares knowledge over time, but with a fixed limit. That is, given the final conclusions, models are preferred in which these conclusions are reached as late as possible. The intuition behind this ordering is similar to the intuition behind $\preceq^g$, and it seems difficult to prefer one to the other on purely intuitive grounds. In Chapter 5, we will see their similarities and differences when used to describe a number of existing forms of reasoning.

Definition 4.47 Define the ordering $\preceq^{gel}$ on TEL-models by

$$\mathcal{N} \preceq^{gel} \mathcal{M} \Leftrightarrow \lim \mathcal{N} = \lim \mathcal{M} \text{ and for all } s \in \mathbb{N} : \mathcal{M}_s \preceq \mathcal{N}_s$$

The letters 'gel' in the notation $\preceq^{gel}$ stand for '*g*lobal, with *e*qual *l*imits'. This ordering can of course be defined on TPL-models in the same way.

The preferential logic based on this ordering again uses conservative models, but only those that are, in addition, closed (see Definition 4.8).

Definition 4.48 (MTEL*)

1. A conservative closed model $\mathcal{M}$ is a $\preceq^{gel}$-*minimal model* of a set of formulae T, denoted $\mathcal{M} \models_{\preceq^{gel}} T$, if
   - $\mathcal{M} \models T$, and
   - for any closed conservative model $\mathcal{N}$, if $\mathcal{N} \models T$ and $\mathcal{N} \preceq^{gel} \mathcal{M}$, then $\mathcal{N} = \mathcal{M}$.
2. A formula φ is a $\preceq^{gel}$-minimal consequence of a set of formulae T, denoted $T \models_{\preceq^{gel}} \varphi$, if $\mathcal{M} \models \varphi$ for every model $\mathcal{M}$ such that $\mathcal{M} \models_{\preceq^{gel}} T$. This preferential logic is called MTEL*.

At this moment, it may be difficult to get a good understanding of these two preference orderings and the effect they have on entailment. In Section 4.5, this will be clarified.

4.4.3 Sequential minimal change

The previous two orderings were based on the intuition that the knowledge of the agent should be minimal, given the fact that it satisfies a description given by temporal formulae. They were used only with conservative models. The last preference criterion we will treat can be used with non-conservative models. The idea is that the *change* of knowledge through time should be minimal. Given a state the agent is in, if a temporal formula prescribes the agent to change its knowledge at the next point in time, it will do so in a minimal fashion. So, first of all we need to establish a notion of minimal change of an information state, and this will depend on the definition of information state. We will treat the epistemic states of IS^{ep} and the partial states in IS^{3val}. Given an information state M, we want to compare other information states on the basis of how similar they are to M. To this end, given M, we introduce an ordering $\leq_M$ on information states where $N_1 \leq_M N_2$ intuitively means that N_1 is more similar to M than N_2.

Definition 4.49

1. Let $M \in \mathsf{IS}^{ep}$. Define an ordering $\leq_M$ on IS^{ep} as follows:

$$N_1 \leq_M N_2 \Leftrightarrow N_1 \triangle M \subseteq N_2 \triangle M$$

   where $\triangle$ denotes the symmetric difference ($A \triangle B = (A \cup B) \setminus (A \cap B)$).

2. Let $m \in \mathsf{IS}^{3val}$. Define an ordering $\leq_m$ on IS^{3val} as follows:

$$n_1 \leq_m n_2 \Leftrightarrow \mathit{Diff}(n_1, m) \subseteq \mathit{Diff}(n_2, m)$$

   where $\mathit{Diff}(m,n) = \mathit{Lit}(m) \triangle \mathit{Lit}(n)$ and $\mathit{Lit}(m) = \{p \mid m(p) = 1\} \cup \{\neg p \mid m(\neg p) = 1\}$.

These orderings enjoy a number of properties.

Proposition 4.50

1. The ordering $\leq_M$ ($\leq_m$) is a partial order.
2. For any $M \in \mathsf{IS}^{ep}$ it holds

$$M \leq_M N \text{ for all } N \in \mathsf{IS}^{ep}.$$

   The same is true for $\leq_m$.

Proof:

1. Reflexivity and transitivity of $\leq_M$ are almost immediate from the definition. Now suppose $N_1 \leq_M N_2$ and $N_2 \leq_M N_1$. Then $N_1 \triangle M = N_2 \triangle M$. We will prove that $N_1 \subseteq N_2$. Let $m \in N_1$ be arbitrary. If $m \notin M$, then $m \in N_1 \triangle M$ so $m \in N_2 \triangle M$. As $m \notin M$, this means $m \in N_2$. If $m \in M$, then $m \notin N_1 \triangle M$ so $m \notin N_2 \triangle M$. As $m \in M$, this means $m \in N_2$. Analogously, $N_2 \subseteq N_1$, so $N_1 = N_2$, proving antisymmetry of $\leq_M$. For $\leq_m$, an analogous proof shows that $\mathit{Lit}(n_1) = \mathit{Lit}(n_2)$ from which $n_1 = n_2$ follows.

2. This is immediate, since $M \bigtriangleup M = \emptyset$ (or, $Lit(m) \bigtriangleup Lit(m) = \emptyset$).

$\square$

The latter property expresses the fact that the information state most similar to M, is M itself.

We now consider models which change as little as possible as time goes on. This means that the information state at time point $t+1$ should be as similar to the one at time point t as possible.

Definition 4.51 Let $\mathcal{M}, \mathcal{N}$ be TEL-models, and let

$$k = \sup\{j \in \mathbb{N} \mid \mathcal{M}_i = \mathcal{N}_i \text{ for all } i < j\}.$$

Define

$$\mathcal{M} \preceq^{sc} \mathcal{N} \Leftrightarrow k = \infty \text{ or } k > 0 \text{ and } \mathcal{M}_k \leq_{\mathcal{M}_{k-1}} \mathcal{N}_k.$$

The definition of this ordering is analogous for TPL-models.

The letters 'sc' in $\preceq^{sc}$ stand for '*s*equential *c*hange'. Based on this preference ordering, we can again define a minimal entailment.

Definition 4.52 (SCTEL)

1. A TEL-model $\mathcal{M}$ is a $\preceq^{sc}$-*minimal model* of a set of formulae T, denoted $\mathcal{M} \models_{\preceq^{sc}} T$, if
   - $\mathcal{M} \models T$, and
   - for any TEL-model $\mathcal{N}$, if $\mathcal{N} \models T$ and $\mathcal{N} \preceq^{sc} \mathcal{M}$, then $\mathcal{N} = \mathcal{M}$.
2. A formula φ is a $\preceq^{sc}$-minimal consequence of a set of formulae T, denoted $T \models_{\preceq^{sc}} \varphi$, if $\mathcal{M} \models \varphi$ for every model $\mathcal{M}$ such that $\mathcal{M} \models_{\preceq^{sc}} T$. This preferential logic is called SCTEL.

4.5 Characterization of minimal models

In this section, we will try to get a better understanding of the notions of minimal entailment defined in Section 4.4. We will do this by studying minimal models of special classes of theories. These theories are sets of formulae that prescribe applying some kind of inference rule. They are generally of the form $\alpha \to \beta$ where α describes some precondition, and β describes the resulting state.

The results are stated and proved for the minimal entailment notions based on TEL, although analogous results are true for the notions based on TPL. First of all, we consider MTEL.

Definition 4.53 (MTEL special formulae)

1. A TEL-formula is called an *input formula* in MTEL, if it of the form $K\gamma_1 \vee \ldots \vee K\gamma_n$ for propositional γ_i. It is called a *reasoning formula* in MTEL, if it is of the following form:

$$(\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \vee \psi_k)$$

   where

$$\varphi_i = K\alpha_1^i \vee \ldots \vee K\alpha_{l(i)}^i \vee \neg FK\beta_1^i \vee \ldots \vee \neg FK\beta_{m(i)}^i$$

   and

$$\psi_i = XK\gamma_i \text{ or } \psi_i = X\neg K\gamma_i$$

   for propositional formulae α_j^i, β_j^i, γ_i, and $n, k, l(i), m(i) \geq 0$.

2. A reasoning formula $(\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \vee \psi_k)$ is *applicable* in a TELC-model $\mathcal{M}$ at point $i \in \mathbb{N}$, if $(\mathcal{M}, i) \models \varphi_1 \wedge \ldots \wedge \varphi_n$.

3. Define $\psi_i' = K\gamma_i$ if $\psi_i = XK\gamma_i$ and $\psi_i' = \neg K\gamma_i$ if $\psi_i = X\neg K\gamma_i$. An S5-model M is said to satisfy the conclusion of a reasoning formula $(\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \vee \psi_k)$ (with φ_i and ψ_i as above) if $M \models \psi_1' \vee \ldots \vee \psi_k'$.

The input formulae describe the initial facts the agent may know. The reasoning formulae describe the effect of applying a rule. The left side of the implication refers to what the agent knows presently, but it may also refer to formulae that the agent will never know in the future. For theories consisting of these formulae, we can give a characterization of their minimal models.

Proposition 4.54 Let T be a theory consisting of input formulae and reasoning formulae of TEL. Then for any conservative TEL-model $\mathcal{M}$, the following holds:

$$\mathcal{M} \models_{\preceq^g} T$$

$$\Leftrightarrow$$

1. $\mathcal{M}_0$ is a $\preceq$-minimal model of the input formulae in T, and

2. For each $i \in \mathbb{N}$, $\mathcal{M}_{i+1}$ is a $\preceq$-minimal element of the set of S5-models which are $\preceq$-extensions of $\mathcal{M}_i$ satisfying the conclusions of the reasoning formulae of T applicable in $\mathcal{M}$ at time point i.

Proof: "$\Leftarrow$": Suppose the above items 1. and 2. hold. We have to prove that $\mathcal{M}$ is a model of T, and then that it is minimal.

• Let $\chi = K\gamma_1 \vee \ldots \vee K\gamma_n$ be an input formula in T. Then $\mathcal{M}_0$ is a model of χ, so $\mathcal{M} \models K\gamma_i$ for some i, and by conservativity, $(\mathcal{M}, t) \models K\gamma_i$ for all $t \in \mathbb{N}$, so $(\mathcal{M}, t) \models \chi$ for all $t \in \mathbb{N}$. Now let $\chi = (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \psi_k)$ with the φ_i, ψ_i as in Definition 4.53. Suppose $(\mathcal{M}, t) \models \varphi_1 \wedge \ldots \wedge \varphi_n$. Then this rule is applicable in $\mathcal{M}$ at t, so $\mathcal{M}_{t+1}$ satisfies its conclusion, which means that there is a ψ_i such that either
- $\psi_i = XK\gamma_i$ and $\mathcal{M}_{t+1} \models K\gamma_i$. Then $(\mathcal{M}, t+1) \models K\gamma_i$ so $(\mathcal{M}, t) \models XK\gamma_i$, whence $(\mathcal{M}, t) \models \psi_1 \vee \ldots \vee \psi_k$.
- $\psi_i = X\neg K\gamma_i$ and $\mathcal{M}_{t+1} \models \neg K\gamma_i$. Then $(\mathcal{M}, t+1) \models \neg K\gamma_i$ so $(\mathcal{M}, t) \models X\neg K\gamma_i$, whence $(\mathcal{M}, t) \models \psi_1 \vee \ldots \vee \psi_k$.

So $\mathcal{M} \models T$.

• Suppose $\mathcal{N}$ is a conservative TEL-model such that $\mathcal{N} \prec^g \mathcal{M}$ and $\mathcal{N} \models T$. Let t_0 be the smallest index for which $\mathcal{N}_{t_0} \prec \mathcal{M}_{t_0}$. Since $\mathcal{N}_0$ must satisfy the input formulae ($\mathcal{N} \models T$), and $\mathcal{M}_0$ is a minimal model satisfying them, it must be the case that $t_0 > 0$. Take a reasoning formula $\chi = (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \vee \psi_k)$ which is applicable at $t_0 - 1$ in $\mathcal{M}$. Then we claim it is applicable in $\mathcal{N}$ at $t_0 - 1$. As χ is applicable in $\mathcal{M}$ at $t_0 - 1$, we have $(\mathcal{M}, t_0 - 1) \models \varphi_1 \wedge \ldots \wedge \varphi_n$. Now take any φ_i. If $(\mathcal{M}, t_0 - 1) \models K\alpha^i_j$ for some j, then, since $\mathcal{M}_{t_0-1} = \mathcal{N}_{t_0-1}$ we have $(\mathcal{N}, t_0 - 1) \models K\alpha^i_j$ so $(\mathcal{N}, t_0 - 1) \models \varphi_i$. Otherwise, it must be the case that $(\mathcal{M}, t_0 - 1) \models \neg FK\beta^i_j$ for some j. As $\mathcal{N} \preceq^g \mathcal{M}$, this means that $(\mathcal{N}, t_0 - 1) \models \neg FK\beta^i_j$, so $(\mathcal{N}, t_0 - 1) \models \varphi_i$. As $\mathcal{N} \models T$, it must be the case that $(\mathcal{N}, t_0 - 1) \models \psi_1 \vee \ldots \vee \psi_k$. But this means that $\mathcal{N}_{t_0}$ is an extension of $\mathcal{N}_{t_0-1} = \mathcal{M}_{t_0-1}$ (conservativity) satisfying the conclusions of formulae applicable in $\mathcal{M}$ at $t_0 - 1$. Since $\mathcal{M}_{t_0}$ is a minimal such extension, it must hold that $\mathcal{M}_{t_0} = \mathcal{N}_{t_0}$, contradicting the definition of t_0. We conclude that such a model $\mathcal{N}$ does not exist, so that $\mathcal{M}$ is a $\preceq^g$-minimal model of T.

"$\Rightarrow$": Suppose $\mathcal{M} \models_{\preceq^g} T$.

• As $\mathcal{M} \models T$, it must be the case that $\mathcal{M}_0$ is a model of the input formulae. Suppose it is not minimal, suppose $N \prec \mathcal{M}_0$ is a model of the input formulae. Define the model $\mathcal{N}$ by $\mathcal{N}_0 = N$ and for $i > 0$: $\mathcal{N}_i = \mathcal{M}_i$. Then obviously $\mathcal{N} \prec^g \mathcal{M}$, and $\mathcal{N}$ is conservative. As N satisfies the input formulae and $\mathcal{N}$ is conservative, $\mathcal{N}$ satisfies the input formulae. Now take a reasoning formulae $\chi = (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \vee \psi_k)$ and suppose for some $t \in \mathbb{N}$ that $(\mathcal{N}, t) \models \varphi_1 \wedge \ldots \wedge \varphi_n$. Take any φ_i; if $(\mathcal{N}, t) \models K\alpha^i_j$ then $(\mathcal{M}, t) \models K\alpha^i_j$. If $(\mathcal{N}, t) \models \neg FK\beta^i_j$, then since $\mathcal{M}_i = \mathcal{N}_i$ for $i > 0$, also $(\mathcal{M}, t) \models \neg FK\beta^i_j$. This means that $(\mathcal{M}, t) \models \psi_1 \vee \ldots \vee \psi_k$. As, again, $\mathcal{M}_i = \mathcal{N}_i$ for $i > 0$, we have that $(\mathcal{N}, t) \models \psi_1 \vee \ldots \vee \psi_k$. Thus, $\mathcal{N}$ is a model of T, in contradiction with $\mathcal{M} \models_{\preceq^g} T$. This means that $\mathcal{M}_0$ is a minimal model of the input formulae.

• Let $i \in \mathbb{N}$. It is easy to see that $\mathcal{M}_{i+1}$ is an extension of $\mathcal{M}_i$ satisfying the conclusions of reasoning formulae applicable in $\mathcal{M}$ at time point i. Suppose it is not minimal, suppose N is smaller. Now define a model $\mathcal{N}$ by $\mathcal{N}_{i+1} = N$ and $\mathcal{N}_j = \mathcal{M}_j$ for $j \neq i$. It is easy to check that $\mathcal{N}$ is a conservative model for which $\mathcal{N} \prec^g \mathcal{M}$. It is also easy to check that $\mathcal{N}$ satisfies the input formulae. So let $\chi = (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow (\psi_1 \vee \ldots \vee \psi_k)$ be a reasoning formula from T. Suppose that $(\mathcal{N}, s) \models \varphi_1 \wedge \ldots \wedge \varphi_n$ for some $s \in \mathbb{N}$. It is straightforward to check that $(\mathcal{M}, s) \models \varphi_1 \wedge \ldots \wedge \varphi_n$; this uses the fact that $\mathcal{N} \preceq^g \mathcal{M}$, that $\mathcal{N}_j = \mathcal{M}_j$ for

$j \neq i$ and conservativity. We will first treat the case when $s \neq i$. This means that $(\mathcal{M}, s) \models \psi_j$ for some j. But since $\mathcal{M}_{s+1} = \mathcal{N}_{s+1}$, we have $(\mathcal{N}, s) \models \psi_j$, so $(\mathcal{N}, s) \models \psi_1 \vee \ldots \vee \psi_k$. Now let us look at the case when $s = i$. Since χ is applicable in $\mathcal{M}$ at time point i, $\mathcal{N}_{i+1} = N$ satisfies the conclusion of χ. If there is a $\psi_j = XK\gamma_j$ such that $N \models K\gamma_j$, then $(\mathcal{N}, i) \models \psi_j$. Otherwise, there is a $\psi_j = X\neg K\gamma_j$ with $N \models \neg K\gamma_j$ so $(\mathcal{N}, i) \models \psi_j$. We have proved that $\mathcal{N} \models T$ which contradicts the fact that $\mathcal{M}$ was a $\preceq^g$-minimal model of T. This means that $\mathcal{M}_{i+1}$ is minimal. □

By restricting the format of the rules further, we get the following characterization.

Corollary 4.55 (Non-disjunctive conclusions) Let T be a theory consisting of input formulae of the form $K\gamma$ and reasoning formulae of the form $\varphi_1 \wedge \ldots \wedge \varphi_n \to \psi$, with the φ_i and ψ as in Definition 4.53. Then for any TELC-model $\mathcal{M}$:

$$\mathcal{M} \models_{\preceq^g} T$$

$$\Leftrightarrow$$

1. $\mathcal{M}_0 = Mod(\{\gamma \mid K\gamma \in T\})$, and
2. For each i, $\mathcal{M}_{i+1} = Mod(Th(\mathcal{M}_i) \cup \{\gamma \mid \text{there is a rule } \varphi_1 \wedge \ldots \wedge \varphi_n \to XK\gamma \in T$ and $(\mathcal{M}, i) \models \varphi_1 \wedge \ldots \wedge \varphi_n\})$, provided $\mathcal{M}_{i+1} \models \neg K\gamma$ for those γ for which there is a rule $\varphi_1 \wedge \ldots \wedge \varphi_n \to X\neg K\gamma \in T$ applicable in $\mathcal{M}$ at time i.

Proof: Using Proposition 4.54, we only have to prove that its conditions 1 and 2 correspond to the above conditions.

1. Let $M = Mod(\{\alpha \mid K\alpha \in T\})$. Then obviously, M is a model of the input formulae. Let N be another model of the input formulae, then $M \preceq N$: take a valuation $m \in N$. Take $K\alpha \in T$ arbitrarily, then since $N \models K\alpha$, we have $m \models \alpha$ so $m \in M$. We have proved that M is the only minimal model of the input formulae, which proves the equivalence of this condition with condition 1 of Proposition 4.54.
2. Let $M = Mod(Th(\mathcal{M}_i) \cup \{\gamma \mid \text{there is a rule } \varphi_1 \wedge \ldots \wedge \varphi_n \to XK\gamma \in T$ and $(\mathcal{M}, i) \models \varphi_1 \wedge \ldots \wedge \varphi_n\})$. Now suppose N is a $\preceq$-minimal $\preceq$-extension of $\mathcal{M}_i$ satisfying the conclusions of the reasoning formulae of T applicable in $\mathcal{M}$ at time point i. Then $M \preceq N$. Take a valuation $m \in N$. Since $\mathcal{M}_i \preceq N$, we have $m \in \mathcal{M}_i$, so $m \models Th(\mathcal{M}_i)$. As N satisfies the conclusions of applicable rules, m satisfies the conclusions of the form $K\gamma$ of applicable rules. Thus, $m \in M$. Furthermore, as N satisfies the conclusions (of applicable rules) of the form $\neg K\gamma$, and $M \preceq N$, also M satisfies them.
   If $\mathcal{M}_{i+1}$ satisfies the condition 2 above, then it obviously satisfies condition 2 of Proposition 4.54. If $\mathcal{M}_{i+1}$ satisfies condition 2 of Proposition 4.54, then it must coincide with M, which then satisfies condition 2 above.

□

If there is no future part in the left hand side of the implications, then minimal models are unique.

Corollary 4.56 Suppose T is a theory as in Corollary 4.55, with the extra restriction that for every rule $\varphi_1 \wedge \ldots \wedge \varphi_n \to \psi$, $\varphi_i = K\alpha_1^i \vee \ldots \vee K\alpha_{l(i)}^i$. If a minimal model of T exists, it is unique.

Proof: Using the characterization of Corollary 4.55, we can prove by induction that for two minimal models $\mathcal{M}$ and $\mathcal{N}$, it must be the case that $\mathcal{M}_i = \mathcal{N}_i$. The crucial observation is that any reasoning formula in T is applicable in $\mathcal{M}$ at point i if and only if this is the case in $\mathcal{N}$. □

So reasoning formulae act as a kind of rules under the semantics of MTEL: an implication of the form $\alpha \to \beta$ can be read as the rule "if α is the case, then conclude β".

For MTEL* (Definition 4.48), we can get a characterization result for a larger class of formulae.

Definition 4.57 (MTEL* special formulae) The MTEL input formulae are also the MTEL* input formulae. Reasoning formulae in MTEL* are also formulae of the form $(\varphi_1 \wedge \ldots \wedge \varphi_n) \to (\psi_1 \vee \ldots \vee \psi_k)$, where the ψ_i are as in Definition 4.53, and

$$\varphi_i = K\alpha_1^i \vee \ldots \vee K\alpha_{l(i)}^i \vee \neg FK\beta_1^i \vee \ldots \vee \neg FK\beta_{m(i)}^i \vee FK\delta_1^i \vee \ldots \vee FK\delta_{n(i)}^i$$

for propositional formulae α_j^i, β_j^i and δ_j^i, and $n, k, l(i), m(i) \geq 0$. The notions of applicability and satisfaction of a conclusion are defined analogously to Definition 4.53.

We again have a characterization result.

Proposition 4.58 Let T be a theory consisting of input formulae and reasoning formulae of MTEL*. Then for any conservative closed TEL-model $\mathcal{M}$, the following holds:

$$\mathcal{M} \models_{\preceq^{gel}} T$$

$$\Leftrightarrow$$

1. $\mathcal{M}_0$ is a $\preceq$-minimal model of the input formulae in T, and
2. For each $i \in \mathbb{N}$, $\mathcal{M}_{i+1}$ is a $\preceq$-minimal element of the set of S5-models which are $\preceq$-extensions of $\mathcal{M}_i$ satisfying the conclusions of the reasoning formulae of T applicable in $\mathcal{M}$ at time point i.

Proof: The proof is completely analogous to the proof of Proposition 4.54. The second part of the "$\Leftarrow$" part of this proof needs the following claim:
If $\mathcal{N} \preceq^{gel} \mathcal{M}$ and $\mathcal{N}_i = \mathcal{M}_i$ for all $i \leq n$, then if a reasoning formula φ is applicable in $\mathcal{M}$ at time point n, it is also applicable in $\mathcal{N}$ at time point n.
First of all, $(\mathcal{M}, n) \models K\alpha \Leftrightarrow (\mathcal{N}, n) \models K\alpha$ since $\mathcal{M}_n = \mathcal{N}_n$. If $(\mathcal{M}, n) \models \neg FK\beta$ then $(\mathcal{N}, n) \models \neg FK\beta$ as $\mathcal{N}_j \preceq \mathcal{M}_j$ for all $j > n$. Lastly, if $(\mathcal{M}, n) \models FK\beta$, then (by closedness!) $\lim \mathcal{M} \models K\beta$, so $\lim \mathcal{N} \models K\beta$, whence (closedness!) $(\mathcal{N}, n) \models FK\beta$.
$\Box$

So under the semantics of MTEL*, an even larger class of formulae act as rules. It also follows from Proposition 4.54 and Proposition 4.58 that MTEL and MTEL* yield the same results (minimal models) on theories containing only input formulae and MTEL reasoning formulae. On MTEL* reasoning formulae, they act differently: consider the formula $FKp \to XKp$. The model $\mathcal{M}$ with $\mathcal{M}_s = \mathrm{Val}(P)$ for all s, is a minimal model in both MTEL and MTEL*. But MTEL* has a second minimal model $\mathcal{N}$ with $\mathcal{N}_0 = \mathrm{Val}(P)$ and $\mathcal{N}_s = Mod(\{p\})$ for $s > 0$.

Corollaries 4.55 and 4.56 also hold for MTEL*.

The third minimality semantics proposed, is SCTEL.

Definition 4.59 (SCTEL special formulae) A SCTEL reasoning formula is a formula of the form $(\varphi_1 \wedge \ldots \wedge \varphi_n) \to (\psi_1 \vee \ldots \vee \psi_k)$ where

$$\varphi_i = K\alpha_1^i \vee \ldots \vee K\alpha_{l(i)}^i \vee \neg K\beta_1^i \vee \ldots \vee \neg K\beta_{m(i)}^i$$

and

$$\psi_i = XK\gamma_i \text{ or } \psi_i = X\neg K\gamma_i$$

for propositional α_i^j, β_i^j, γ_i, and $n, k, l(i), m(i) \geq 0$. The notions of applicability and satisfaction of a conclusions is again defined analogously to Definition 4.53.

Proposition 4.60 Let T be a theory containing only SCTEL reasoning formulae. Suppose that for every $S \subseteq T$ the following holds: if $\{\varphi_1 \wedge \ldots \wedge \varphi_n \mid (\varphi_1 \wedge \ldots \wedge \varphi_n) \to (\psi_1 \vee \ldots \vee \psi_k) \in S\}$ is S5-satisfiable, then $\{\psi_1' \vee \ldots \vee \psi_k' \mid (\varphi_1 \wedge \ldots \wedge \varphi_n) \to (\psi_1 \vee \ldots \vee \psi_k) \in S\}$ is S5-satisfiable. Then for any TEL-model $\mathcal{M}$, the following holds:

$$\mathcal{M} \models_{\preceq^{sc}} T$$

$$\Leftrightarrow$$

For each $i \in \mathbb{N}$, $\mathcal{M}_{i+1}$ is an S5-model which is $\leq_{\mathcal{M}_i}$-minimal among models satisfying the conclusions of the reasoning formulae of T applicable in $\mathcal{M}$ at time point i.

Proof: "$\Rightarrow$": Suppose $\mathcal{M} \models_{\preceq^{sc}} T$, but it is not the case that $\mathcal{M}_{i+1}$ is minimal. Then there is an S5-model N such that $N \leq_{\mathcal{M}_i} \mathcal{M}_{i+1}$ satisfying the applicable rules. Now define a TEL-model $\mathcal{N}$ as follows. For $j \leq i$, set $\mathcal{N}_j = \mathcal{M}_j$, and let $\mathcal{N}_{i+1} = N$. Now take the set of rules $(\varphi_1 \wedge \ldots \wedge \varphi_n) \to (\psi_1 \vee \ldots \vee \psi_k)$ for which $N \models \varphi_1 \wedge \ldots \wedge \varphi_n$. This set is obviously satisfiable, so the assumption on T implies that the set $\psi'_1 \vee \ldots \vee \psi'_k$ is S5-satisfiable. Let $\mathcal{N}_{i+2}$ be any S5-model satisfying $\psi'_1 \vee \ldots \vee \psi'_k$. Now we can again look at the set of rules whose left-hand side is applicable in $\mathcal{N}_{i+2}$. This set is satisfiable, so we can find a suitable $\mathcal{N}_{i+3}$, etcetera. It is easy to check that the resulting model $\mathcal{N}$ is a model of T, and furthermore, that $\mathcal{N} \preceq^{sc} \mathcal{M}$. This is in contradiction with the assumption that $\mathcal{M}$ was a $\preceq^{sc}$-minimal model of T, so $\mathcal{M}_{i+1}$ must satisfy the right-hand side of the equivalence.
"$\Leftarrow$": Suppose we have a TEL-model $\mathcal{M}$ for which the right-hand side of the equivalence holds. It is straightforward to check that $\mathcal{M} \models T$. Now suppose there exists a model $\mathcal{N}$ of T with $\mathcal{N} \preceq^{sc} \mathcal{M}$ and $\mathcal{N} \neq \mathcal{M}$. Then there is an index k such that $\mathcal{N}_j = \mathcal{M}_j$ for $j < k$, and $\mathcal{N}_k \leq_{\mathcal{M}_{k-1}} \mathcal{M}_k$ (and $\mathcal{N}_k \neq \mathcal{M}_k$). But obviously $\mathcal{N}_k$ satisfies the formulae $\psi'_1 \vee \ldots \vee \psi'_k$ for the rules $(\varphi_1 \wedge \ldots \wedge \varphi_n) \to (\psi_1 \vee \ldots \vee \psi_k)$ such that $\mathcal{M}_{k-1} = \mathcal{N}_{k-1} \models \varphi_1 \wedge \ldots \wedge \varphi_n$, contradicting the assumption. Thus, $\mathcal{M}$ is minimal and $\mathcal{M} \models_{\preceq^{sc}} T$. □

For future use (Section 8.1) we give a similar characterization result for temporal partial logic. The admissible formulae are shifted one step in time (instead of 'present implies next', they are 'yesterday implies current'), and allow looking back one extra point in time. The notions of 'applicable formula', and 'satisfying the conclusions of applicable rules' are defined as would be expected.

Proposition 4.61 Let T be a set of formulae of temporal partial logic of the form $\bigwedge \Psi \to \varphi$, where Ψ only contains formulae of the form $YYCk$, $\neg YYCk$, YCk, and $\neg YCk$, where k is a literal, and φ is either Ck (k again a literal) or $\neg Ca \wedge \neg C\neg a$ (for a an atom). Furthermore, suppose that for any set $S \subseteq T$, if $\bigwedge\{\bigwedge \Psi \mid (\bigwedge \Psi \to \varphi) \in S\}$ is satisfiable, then $\bigwedge\{\varphi \mid (\bigwedge \Psi \to \varphi) \in S\}$ is satisfiable. Then the following statements are equivalent for every TPL-model $\mathcal{M}$:

1. $\mathcal{M}$ is a $\preceq^{sc}$-minimal model of T;
2. $(\mathcal{M}, 0) \models T$, and for each $i \in \mathbb{N}$, the partial model $\mathcal{M}_{i+1}$ is $\leq_{\mathcal{M}_i}$-minimal among all models satisfying the conclusions of the rules applicable at time point $i+1$;
3. $(\mathcal{M}, 0) \models T$, and for each literal k and $i \in \mathbb{N}$, we have that k is true in $\mathcal{M}_{i+1}$ if and only if either
   - there is a formula $\bigwedge \Psi \to Ck$ in T such that $(\mathcal{M}, i+1) \models \bigwedge \Psi$, or
   - there is no formula $\bigwedge \Psi \to C\neg k$ (if $k = \neg a$ then read $C\neg k$ as Ca) or $\bigwedge \Psi \to \neg Ca \wedge \neg C\neg a$ (if k is either a or $\neg a$) with $(\mathcal{M}, i+1) \models \bigwedge \Psi$, and k is true in $\mathcal{M}_i$.

Proof: The proof of equivalence of the first two statements is completely analogous to the proof of Proposition 4.60; the formulae here are just shifted one point in time and the extra $(\neg)YYCl$ conjuncts change nothing in the proof.

Now the equivalence between the second and third statement. If we consider $\mathcal{M}$ and a time point i, then there are a number of rules applicable at time $i+1$. Now take $\mathcal{M}_i$ and change a literal if there is an applicable rule changing it. Call the result M. It is easy to see that this M is exactly the (unique) model satisfying the description in the third statement. It always exists since the applicable rules form (by definition) a satisfiable set of left hand sides, and therefore their right hand sides are not inconsistent (by the assumption in the proposition). It is also straightforward that M is the only model $\preceq^{sc}$-minimal among models satisfying right hand sides of applicable rules. □

The third statement in the above proposition opens the way to a completion procedure: the exact conditions for a literal to be true can be described in temporal partial logic, using the *monotonic*, classical semantics of TPL. This means that for any theory T, we can form its completion T', such that the $\preceq^{sc}$-minimal models of T are the (classical) models of T'. Then we can use classical entailment on T' to find the SCTEL-consequences of T. We will describe this completion technique (for slightly more general theories) in Section 8.1.

The minimal change criterion gives belief revision complying with the AGM postulates [AGM85].

Proposition 4.62

1. *Expansion:* Suppose $\mathcal{M}_0 \not\models K\neg\alpha$. Then $\mathcal{M} \models_{\preceq^{sc}} XK\alpha \Leftrightarrow \mathcal{M}_j = \mathcal{M}_0 \setminus \{m \in \mathcal{M}_0 \mid m \not\models \alpha\}\ \forall j > 0$. If $\mathcal{M} \models_{\preceq^{sc}} XK\alpha$, then $Th(\mathcal{M}_1) = Cn(Th(\mathcal{M}_0 \cup \{\alpha\})$.

2. *Contraction:* $\mathcal{M} \models_{\preceq^{sc}} X\neg K\alpha \Leftrightarrow \begin{cases} \mathcal{M}_j = \mathcal{M}_0\ \forall j & \text{if } \mathcal{M}_0 \models \neg K\alpha \\ \mathcal{M}_j = \mathcal{M}_0 \cup \{m\} & \text{for some } m \text{ such that} \\ & m \not\models \alpha \text{ otherwise.} \end{cases}$
   If $\mathcal{M} \models_{\preceq^{sc}} X\neg K\alpha$, then $Th(\mathcal{M}_1)$ is a maximal deductively closed subset of $Th(\mathcal{M}_0) \setminus \{\alpha\}$.

3. *Revision:* Suppose $\mathcal{M}_0 \models K\neg\alpha$ and $\mathcal{M} \models_{\preceq^{sc}} XK\alpha$. Then $\mathcal{M}_j = \{m\}$ for some m such that $m \models \alpha$ (for all $j > 0$).

Proof:

1. "$\Rightarrow$": Define $\mathcal{N}_0 = \mathcal{M}_0$ and $\mathcal{N}_j = \mathcal{N}_0 \setminus \{m \in \mathcal{N}_0 \mid m \not\models \alpha\}$ for $j > 0$. Then obviously $\mathcal{N} \models XK\alpha$. Furthermore, $\mathcal{N}_1 \leq_{\mathcal{M}_0} \mathcal{M}_1$: let $w \in \mathcal{N}_1 \triangle \mathcal{M}_0$, then $w \in \mathcal{M}_0 \setminus \mathcal{N}_1$ so $w \not\models \alpha$ so $w \notin \mathcal{M}_1$ as $\mathcal{M}_1 \models K\alpha$. Thus we have $w \in \mathcal{M}_1 \triangle \mathcal{M}_0$. But as $\mathcal{M} \models_{\preceq^{sc}} XK\alpha$, it must be the case that $\mathcal{M}_1 = \mathcal{N}_1$. For $j > 1$ the statement is now easy to prove.
   "$\Leftarrow$": Analogous and straightforward.

Suppose $\mathcal{M}_1 = \{m \in \mathcal{M}_0 \mid m \models \alpha\}$. We have to prove that $Th(\mathcal{M}_1) = Cn(Th(\mathcal{M}_0 \cup \{\alpha\})$.
"$\supseteq$": Suppose $Th(\mathcal{M}_0) \cup \{\alpha\} \models \beta$. Let $m \in \mathcal{M}_1$, then $m \in \mathcal{M}_0$ so $m \models Th(\mathcal{M}_0)$ and $m \models \alpha$. But then $m \models \beta$. We have that $\beta \in Th(\mathcal{M}_1)$.
"$\subseteq$": Suppose $\beta \in Th(\mathcal{M}_1)$, so $\forall m \in \mathcal{M}_0 : (m \models \alpha \Rightarrow m \models \beta)$ so $\forall m \in \mathcal{M}_0 : m \models \alpha \to \beta$ so $\alpha \to \beta \in Th(\mathcal{M}_0)$ so $\beta \in Cn(Th(\mathcal{M}_0) \cup \{\alpha\})$.

2. The case when $\mathcal{M}_0 \models \neg K\alpha$ is easy, so suppose $\mathcal{M}_0 \models K\alpha$.
"$\Rightarrow$": Suppose $\mathcal{M} \models_{leqltr} \neg K\alpha$, then there exists $m \in \mathcal{M}_1 \setminus \mathcal{M}_0$ such that $m \not\models \alpha$. Define $\mathcal{N}_0 = \mathcal{M}_0$, $\mathcal{N}_j = \mathcal{M}_0 \cup \{m\}$ for $j > 0$. It is easy to see that $\mathcal{N} \models \neg K\alpha$. Furthermore, $\mathcal{N}_1 \leq_{\mathcal{M}_0} \mathcal{M}_1$: take $w \in \mathcal{N}_1 \triangle \mathcal{M}_0$, then $w = m \in \mathcal{M}_1 \setminus \mathcal{M}_0 \subseteq \mathcal{M}_1 \triangle \mathcal{M}_0$. But $\mathcal{M} \models_{leqltr} \neg K\alpha$, so $\mathcal{N}_1 = \mathcal{M}1$. The statement for $j > 1$ is easy.
"$\Leftarrow$": straightforward.

   Now suppose $\mathcal{M} \models_{\preceq^{sc}} X\neg K\alpha$. If $\mathcal{M}_0 \not\models K\alpha$, then it is straightforward to check that $Th(\mathcal{M}_1)$ is a maximal deductively closed subset of $Th(\mathcal{M}_0) \setminus \{\alpha\}$, since $Th(\mathcal{M}_1) = Th(\mathcal{M}_0) \setminus \{\alpha\}$. So suppose $\mathcal{M}_0 \models K\alpha$. Then we know that $\mathcal{M}_j = \mathcal{M}_0 \cup \{m\}$ for some m such that $m \not\models \alpha$.
   • Suppose $\mathcal{M}_1 \models \beta$, then $\mathcal{M}_0 \models \beta$, so $\beta \in Th(\mathcal{M}_0)$ and $\beta \neq \alpha$ since $\mathcal{M}_1 \not\models \alpha$.
   • Obviously, $Th(\mathcal{M}_1)$ is deductively closed.
   • Maximality: Suppose there is a deductively closed set of formulae T with $Th(\mathcal{M}_1) \subsetneq T \subseteq Th(\mathcal{M}_0) \setminus \{\alpha\}$. Then there is a $\beta \in T$ such that $\beta \notin Th(\mathcal{M}_1)$. But $\alpha \neq \beta$ so $\beta \in Th(\mathcal{M}_0)$. We claim that $\beta \to \alpha \in Th(\mathcal{M}_1)$: choose $k \in \mathcal{M}_1$: if $k \in \mathcal{M}_0$ then $k \models \alpha$ so $k \models \beta \to \alpha$. If not, then $k = m$. As $\beta \in Th(\mathcal{M}_0)$ and $\beta \notin Th(\mathcal{M}_1)$, it must be the case that $m \not\models \beta$. But that implies $m \models \beta \to \alpha$. As $\beta \to \alpha \in Th(\mathcal{M}_1) \subset T$, we have $\beta \to \alpha \in T$ and with $\beta \in T$ we get $\alpha \in T$, in contradiction with our assumption. This means such a T can not exist.

3. Suppose $\mathcal{M}_0 \models K\neg\alpha$ and $\mathcal{M} \models_{\preceq^{sc}} XK\alpha$. Define $\mathcal{N}_0 = \mathcal{M}_0$ and $\mathcal{N}_j = \{m\}$ for an arbitrary $m \in \mathcal{M}_1$ (for $j > 0$). Then $\mathcal{N} \models XK\alpha$, and $\mathcal{N}_1 \leq_{\mathcal{M}_0} \mathcal{M}_1$: take $k \in \mathcal{N}_1 \triangle \mathcal{M}_0$. If $k \in \mathcal{N}_1 \setminus \mathcal{M}_0$, then $k = m \in \mathcal{M}_1 \setminus \mathcal{M}_0$. If $k \in \mathcal{M}_0 \setminus \mathcal{N}_1$ then $k \models \neg\alpha$ so $k \notin \mathcal{M}_1$ so $k \in \mathcal{M}_0 \setminus \mathcal{M}_1$. This implies, with $\mathcal{M} \models_{\preceq^{sc}} XK\alpha$, that $\mathcal{M}_1 = \mathcal{N}_1$. The statement for $j > 1$ is straightforward.

□

The above proposition implies that belief revision induced by the minimal change criterion satisfies the AGM postulates. Expansion of a theory by a formula α can only be done by adding α and closing the result under Cn [Gär92b]. The second item in the above proposition shows that contraction is done via a "maxichoice contraction function" (deleting α from a theory yields a maximal deductively closed subset not containing α; see [AGM85]), which implies (by a result from [AGM85]) that the postulates are satisfied. From the third item of Proposition 4.62 it is straightforward to show that revision of a theory by a formula α in SCTEL can be seen as contraction of the theory by $\neg\alpha$, followed by expansion by α. The Levi-identity ensures that the resulting revision satisfies the AGM postulates (see [AGM85]).

4.6 Temporal logic as a specification language

In the previous sections, a number of temporal languages (TEL and TPL) and a number of notions of modelhood of a theory ($\models$, $\models^c$, $\models_{\preceq^g}$, $\models_{\preceq^{gel}}$ and $\models_{\preceq^{sc}}$) were defined, and it was suggested that these temporal logics could be used as specification languages. But that means that a temporal theory should give rise to a reasoning frame operator, analogously with the reasoning frame operator definition for default logic and logic programming of Sections 3.1 and 3.2. A temporal theory, together with a notion of modelhood, gives rise to a set of temporal models (the set of models of the theory). These temporal models are sequences of information states, and are thus traces. In order to define a reasoning frame operator, we can assign to a set of initial facts the set of all temporal models of the theory whose first state (in the sequence) contains exactly these initial facts.

Definition 4.63 (Reasoning frame operator of a theory) Let S be a set of formulae of TEL (or TPL), and let $\Vdash$ be a relation of modelhood between a class of temporal models $\mathcal{C}$ and formulae of TEL (respectively, TPL). The reasoning frame operator *specified by* S, denoted $\mathcal{T}_S$, is defined by

$$\mathcal{T}_S(X) = \{\mathcal{M} \in \mathcal{C} \mid \mathcal{M} \Vdash S \text{ and } Th(\mathcal{M}_0) = Cn(X)\}$$

for all sets of propositional formulae X.

Although temporal logic induces reasoning frame operators in a very natural manner, it can also be used to specify multiple belief state operators using the techniques of Chapter 2 (Definition 2.25).

Definition 4.64 (MBSO of a theory) Let S be a set of formulae of TEL (or TPL), and let $\Vdash$ be a relation of modelhood between a class of temporal models $\mathcal{C}$ and formulae of TEL (respectively, TPL). The MBSO *specified by* S, denoted $\mathcal{B}_S$, is defined by

$$\mathcal{B}_S(X) = \{\lim\ \mathcal{M} \in \mathcal{C} \mid \mathcal{M} \Vdash S \text{ and } Th(\mathcal{M}_0) = Cn(X)\}$$

for all sets of propositional formulae X.

Of course, using the definitions of Chapter 2, we can also define an associated FBSO.

In the next chapter, some example theories are given that specify various forms of reasoning in this way.

4.7 Conclusions

In this chapter we have defined a number of temporal logics of information. These logics can serve to describe the changes over time of the information of a rational agent. There are many choices that can be made for the design of these logics, yielding many different logics. First of all, choices have to be made regarding the nature of a state of information. This can be a Kripke model, or a partial model, or we can draw from any other information state frame of Chapter 2. One option we would like to explore in the future is to use an epistemic modal logic different from S5. One may not want to impose the strong principles of (positive and negative) introspection on the knowledge of the agent, so one could use S4 or T instead. Also, approaches based on the distinction between implicit and explicit belief could be used (see for instance [Lev84]; [FHMV95] contains an overview), where implicit knowledge satisfies the S5 axioms, but the explicit knowledge need not even be closed under logical consequence.

Another interesting extension of the logics described here is the addition of an 'actual' world. The logics of this chapter model the change of belief only; there is no changing *real world*. The S5-models at a point in time do not have an actual world (and indeed, we have taken a fragment of S5, containing only the subjective formulae, which does not refer to the real world). If we define an information state to be a pair consisting of an S5-model, together with a propositional model (this yields a KD45 formalization; if the propositional model must be an element of the S5-model, we get S5), and we base the temporal language on the full S5-language (not just the subjective part), then we could model the change of the world and the agent's beliefs about this world. One of the reasons we have not done this, is that we wanted to focus on the agent's beliefs proper first. This new variant would allow us to model observation and communication of the agent with the world. For example, if the agent wants to observe whether p holds in the world, this could be described by the formulae $p \to X(Kp)$ and $\neg p \to X(K\neg p)$, where the observation takes one step in time.

Third, the ontology of time can be varied. In this chapter, we have devoted attention to a very simple structure of time, the natural numbers, and a branching time variant. But there are other possibilities. We could drop the assumption that there is a starting point in time. It may also be useful to consider dense time, which may be used to model the gradual formation of beliefs, but which is also needed if there is an outside world that is modeled, in which dense time is called for (for example if there are other agents or events that are not necessarily synchronized). But even circular time can be thought of (for instance for an agent that needs to think the same things every week). Of course also the interaction between the information states and time can be varied. The property of conservativity is an example of such an interaction, which is sometimes useful, and sometimes is not.

Lastly, even with the same semantical objects, the language can be varied. For the temporal part, we have taken a standard tense-logical language. We could also define a Since and Until operator. In the case of branching time, we could incorporate all the CTL*-operators (instead of only the CTL-operators). The language we defined did not allow us to reason about the knowledge the agent has about time (a formula like $K(Kp \rightarrow G(Kp))$, expressing the fact that the agent knows that its belief of p will persist in time, is not in our language). It would be interesting to allow the agent to reason about its own reasoning in time. Such choices affect the expressiveness and the complexity of the logic.

We defined a number of semantical 'filters' to obtain *preferred* (or *intended*) models of a formula, by defining orderings on temporal models, and by considering only minimal models of a formula according to these orderings (notice that this model-selection actually defines an FBSO according to Example 2.8: reasoning about reasoning is of course also a form of reasoning, subject to the same considerations of Chapters 1 and 2). A number of characterization results were given in order to clarify the behavior of the different forms of minimal entailment.

One extension we have not mentioned before is the extension to more than one agent. This would allow the description of the progression of a complete multi-agent system over time. In Chapter 8.1, an extension of temporal partial logic to more agents is described, but this is a 'flat' extension, in which beliefs are never nested: an agent can not reason about the beliefs of other agents. In principle, such an extension would be easy: we can take S5-models for more than one agent (sometimes called $S5_{(n)}$-models), and a language with a different K-operator for every agent. We could even take more extended formalizations of multiple agents, taking into account knowledge, belief, actions, abilities, opportunities, desires, intentions, etc. (see for instance [Lin96], [RG92] or [Sin94]). The difficulties arise in the generalization of the preference orderings (more generally, in the definition of the information ordering). We already mentioned the fact that MTEL (and MTEL*) can be seen as temporalizations of the logic Ground S5 of 'only knowing' (see [HM85b]; Ground S5 is defined formally in Section 9.2.2). This approach has been generalized to the case of more than one agent (see [Hal97]), and these results can probably be used to define a version of MTEL for many agents.

4.8 Related work

Temporal logic per se has a long tradition, and we refer the interested reader to, for instance, [Ben91b] or [Gol92]. In theoretical computer science, temporal logic has been used to specify and reason about programs (or processes) (see [Kro87], [Eme90] and [MP92]). The difference with our logics is mainly in the information states: for processes, these consist of the values of program (process) variables, whereas our states contain knowledge of the agent. Knowledge of processes is modeled by the approach of Halpern and Moses ([HM90], see also [FHMV95]). This is based on the notion of a *run* of a system consisting of a number of processes (and an environment), which is basically a sequence of states, where a state is a tuple of the local states of the processes. The epistemic accessibility relation is fixed as soon as we have a set of runs: a process i at a given point (r, m) (where r is a run and m is a natural number

— the time point) considers possible all points (r', m') where its local state is the same as in (r, m). So the notion of a state is central, and the knowledge of a process is derived from it, in contrast to our logics, where a state is uniquely determined by the knowledge contained in it. This can be modeled in their approach, however, as shown in [FHMV95], by defining a state to be a set of (known) propositions.

Another logic of belief over time is the temporal belief logic (TBL) of [FW97]. It has the same model of time (the natural numbers) as MTEL. However, the states are essentially syntactical: at any point in time, a set of propositions is associated to every agent (there may be more than one agent). This set is the set of propositions the agent believes at that point in time. The logic is used to reason about systems specified in METATEM, which can alternatively be seen as the executable version of TBL. We will come back to METATEM after discussing the executability of our temporal logics (in Chapter 6).

The logic TEMA (Temporal Epistemic Meta-level Architecture) of [HMT94] was designed specifically to formalize the reasoning process of a meta-level architecture. It is a temporalization (using labeled branching time structures) of S5P*, a modal epistemic logic which contains, apart from a knowledge and a belief operator, a specific class of operators that denote possible assumptions. The formula $P_i\varphi$, for example, means that φ is a possible assumption within frame of mind i. In temporal S5P*-models that obey “downward reflection”, all possible assumptions are added to the beliefs of the agent at some next point(s) in the future. The class of S5P*-models forms an information state frame, where the ordering reflects an increase in *belief.* As such, the approach of [HMT94] falls within the general framework described in this chapter. One of the problems with temporal S5P*-models is that there may be many unintended models, in which the agent may form new beliefs without good reason: the agent may come to belief α, although there were no new possible assumptions previously that imply α. In a meta-level architecture, new object-level beliefs only arise through assumptions at the meta-level (excluding derived beliefs and direct observations). Using a minimality preference ordering such as the ones introduced in this chapter may eliminate these unintended models.

There are a number of other logics in which the beliefs of agents over time can be described, among which the already mentioned [RG92] and [Sin94]. In the presence of these ‘powerful’ logics, it may seem that MTEL is useless. However, the high expressivity of the above logics comes with some logical drawbacks: many of them do not have a known complete axiomatization, and they have a very high (computational, if not conceptual) complexity. This is in contrast to the nice logical properties of TEL (See Section 9.1). Furthermore, we will see in Chapter 5 that many forms of nonmonotonic reasoning can be described in TEL augmented with a preference ordering. This is not (directly) possible in any of the above logics (and they were indeed not defined for this purpose, in contrast to MTEL).

There are a number of approaches that are intended to model the effect of incoming information on the information state of an agent, called dynamic semantics or update semantics ([Gro95, Vel96]; for a more general discussion of logical dynamics, see [Ben96a]). The basic idea is that every formula has an associated action which has an effect on information states (for example incorporating the belief in that formula in the information state of an agent): the semantics of such a formula is the change of information state its action induces. In some of these approaches, one could have a formula of the form $[\varphi]\psi$, which means that ψ is true whenever the agent learns the information in φ. Such approaches could be called dynamic logics of information and the difference with our temporal logics of information is like the difference between dynamic logic and temporal logic. The most important differences are that: (1) actions are not explicitly modeled or specified in temporal logic, and (2) the history of change is not explicitly modeled in dynamic logic. The meaning of the formula $[\varphi]\psi$ implicitly incorporates an assumption of minimality: when updating with φ, the agent does not learn anything besides (not caused by) φ. This is not the case in our non-minimal temporal logics (TEL, TELC, and TPLC), where a temporal model may contain knowledge which is not specified by a formula it satisfies: there are models of, for instance, the formula Kp, in which also at some point in time Kq holds. When using minimal entailment, this does not occur (the precise details depend in the precise ordering): the $\preceq^g$-minimal model of Kp has no time point at which Kq holds (or any other formula not implied by p). There are also some variants of dynamic semantics where entailment of the form $\varphi_1, \ldots, \varphi_n \models \psi$ is investigated, where the order of $\varphi_1, \ldots, \varphi_n$ is important: that is the order in which the agent receives new information. This defines in a sense a trace of information states, where each new state in the trace is induced by the new information. A similarity between our logics and many of the above dynamic approaches, is that the information states are often modal models. Sometimes they are just sets of propositional valuations (just as our normal S5-models; this is the case in [Vel96]), sometimes they are modal partial models (see [Jas94]), but they can also be 'increasing' sequences of structures containing higher-order beliefs about other agents (as described in [Gro95]). There may be many interesting connections between dynamic and temporal approaches to modeling information change; we leave that for further research.

There are two other approaches that are based on ideas about the importance of *dynamic aspects* of reasoning that are very similar to our own, namely step-logic (or, more generally, active logics, see [Elg88] or [EP90]) and the situation calculus semantics for logic programs (see [LR96]). We will not discuss these approaches here, as it will be more convenient to do so at the end of Chapter 5.

One field of research that has to be mentioned here is nonmonotonic temporal reasoning, which is concerned with commonsense reasoning about actions and change in the world (see [SS95] for an overview). In particular the approach of *chronological minimization* (see [Sho88]) is quite similar to our own: there we also see temporal models, with a preference relation to select the intended models. The logic SCTEL, based on the ordering $\preceq^{sc}$ formalizes the same intuition that everything stays the same as much as possible: a successor state is preferred if it is more similar to the current state. In a sense, both our and Shoham's approach try to minimize change, Shoham's approach geared towards the physical world, our approach meant for the internal epistemic state of an agent. This also dictates the difference in what a state is. The other preference relations (which will be used more extensively than $\preceq^{sc}$ in the rest of this thesis), do not immediately seem to have a meaningful equivalent in the physical world. Later, different variants of Shoham's ordering have been proposed (see [SS95]).

Acknowledgments

Temporal partial logic, and the $\preceq^{g}$-ordering (MTPL), first appeared in [ET93]. A branching time variant of minimal temporal partial logic was used in [ET94b]. The logic MTEL appeared in [Eng96a], and a branching time variant was introduced in [ET96b]. The special formulae for MTEL are a generalization of the type of formulae studied in [ET96c] and [ET96a], and their counterparts in MTPL are used in [EH97]. The material of Section 4.3 is part of [ET94a].

Chapter 5

Reasoning Processes in Temporal Logic

In the previous chapter, a number of variants of temporal logic were introduced and proposed as a specification language for reasoning processes. In this chapter, we give some examples of forms of reasoning which can be specified in temporal logic. The reasoning frame operators thus specified will be given explicitly wherever appropriate.

5.1 Default logic: the linear case

In this section, we will show how the reasoning behavior of an agent reasoning with default logic can be specified in temporal logic. As we mentioned in Section 3.1, default reasoning has an essential temporal element. To be more specific, as the agent is reasoning she has to assess a specific type of condition (the justification) in default rules to be applied, that cannot be fulfilled only on the basis of what has been derived until that moment. After application of a default rule, only in the *future of the reasoning process* can it be verified whether this condition of the applied default rule was justified or unjustified. This suggests that a default rule can be given an interpretation as a temporal rule with one of its conditions referring to the future of the reasoning process. It seems that the process of actually constructing a set of coherent assumptions is reflected in Reiter's approach to a certain extent, but without making the essential temporal element explicit.

As the translation into temporal logic yields a temporal semantics, this indirectly gives semantics to default logic. The main contribution of this semantics for default logic when compared to other approaches is that the temporal element in default reasoning is made explicit in the semantics. This enables better insight in the inherent dynamic nature of default reasoning. In a sense, the temporal rules prescribe the agent to perform the action of applying a default rule.

The temporal interpretation has two other benefits besides giving a natural semantics to default logic. The first one is that we can express and reason about temporal properties of the reasoning process. It may, for instance, be interesting to know which conclusions can be drawn before a certain point in time. Secondly, it is possible to use classical temporal logic to reason about default theories. Even though the translation is into MTEL, which uses a minimization of models, in general it may be possible to derive interesting properties even without minimization.

The next section describes the translation of a default theory into a temporal theory.

5.1.1 A temporal interpretation of default logic

Let $\Delta = \langle D, W \rangle$ be a default theory. A trace of a default reasoning process based on Δ is described by a sequence of epistemic states with increasing information (as formalized in Section 3.1), and the semantics of our temporal logics are based on these traces. The translation of Δ into temporal logic should therefore specify which properties a default reasoning temporal model should have. The initial state of a default reasoning temporal model $\mathcal{M}$ (and therefore all subsequent states as well) includes the knowledge in W; therefore $\mathcal{M}$ should be a model of $\{K\alpha \mid \alpha \in W\}$.

Suppose a default rule $(\alpha : \beta_1, \ldots, \beta_n)/\gamma$ is given with propositional formulae α, $\beta_1, \ldots, \beta_n$, γ and α is known to be true in $\mathcal{M}$ at time point t; i.e., $(\mathcal{M}, t) \models K\alpha$. If the default rule is applicable then its consequent is required to be true in the next state (and by conservativity in all subsequent states); i.e., $(M, t) \models XK\gamma$. What remains is how to express whether application of the default rule is justified. The requirement is that the β_i are consistent in the context of the reasoning process, including the part of the context yet to be generated by further reasoning steps. Since the reasoning is conservative, this means that there should be no future state where $\neg\beta_i$ is generated for any i. In the temporal logic we designed this is quite easy to express: it is required that $(\mathcal{M}, t) \models \neg FK(\neg\beta_i)$. If we compare this to the translation of the justification for nonmonotonic modal logics as defined in [MST93], $LM\beta_i$ (the agent should know that it considers β_i possible), one sees that our translation is the dynamic variant of their translation. The agent considers β_i possible just in case she never derives $\neg\beta_i$. Summarizing, for our temporal model we require: if α is known to be true at time point t and for no i is $\neg\beta_i$ known to be true at any time point after t, then γ should be known to be true at the time point $t + 1$. Note that, as S5 coincides with KD45 on subjective formulae, one can read 'believes' for 'knows' in the above discussion. In temporal epistemic logic, this translates into the formula:

$$K\alpha \wedge \neg FK(\neg\beta_1) \wedge \ldots \wedge \neg FK(\neg\beta_n) \rightarrow XK\gamma.$$

This leads us to the following definition:

Definition 5.1 (Temporal interpretation mapping for default theories)
Let $\Delta = \langle D, W \rangle$ be a default theory.

1. The mapping τ, associating with any default rule $(\alpha : \beta_1, \dots, \beta_n)/\gamma$ its *temporal interpretation*, is defined by

$$\tau : (\alpha : \beta_1, \dots, \beta_n)/\gamma \mapsto K\alpha \wedge \neg FK(\neg\beta_1) \wedge \dots \wedge \neg FK(\neg\beta_n) \rightarrow XK\gamma.$$

   The set $\tau(D)$ is called the *temporal interpretation* of the set of default rules D.

2. The *temporal interpretation* of W is defined by $\tau(W) = \{K\alpha \mid \alpha \in W\}$.

3. The *temporal interpretation of the default theory* Δ is defined by:

$$\tau(\Delta) = \tau(D) \cup \tau(W).$$

The temporal interpretation of Δ ensures that any default rule which is applicable, is actually applied. However, we also want to make sure that these default conclusions are *the only ones* which are added to the knowledge of the reasoner. As will be seen in the next section, this can be accomplished by taking the minimal temporal models of the interpretation of the default theory, with respect to the ordering $\preceq^g$ defined in Section 4.4.

5.1.2 Semantical correspondences

In the previous section we defined a correspondence between (sets of) default rules and (sets of) temporal formulae at a syntactic level, and we gave an informal sketch of the semantics behind this syntactic translation. In this section we will give a formal treatment of the related semantical correspondence between Reiter extensions and $\preceq^g$-minimal temporal models, induced by the interpretation mapping τ.

The semantical correspondence will be such that the minimal temporal models of the temporal theory are the traces of the extensions of the default theory. This means that if the temporal model $\mathcal{M}$ corresponds to the extension E (with the sets E_i defined as in Definition 3.1), then $\mathcal{M}_i$ contains the knowledge of E_i and the limit $\lim \mathcal{M}$ contains the knowledge of E:

$$\begin{array}{ll} \mathcal{M}_t = Mod(E_t) & E_t = \mathrm{Th}(\mathcal{M}_t) \\ \lim \mathcal{M} = Mod(E) & E = \mathrm{Th}(\lim \mathcal{M}). \end{array}$$

Before proving this correspondence, we give an example.

Example 5.2 Let $\Delta = \langle D, W\rangle$ be a default theory in the language with $P = \{a, b, c, d, e\}$, defined by

$$\begin{array}{rcl} D & = & \{(a : b)/b, (d : c)/c, (b : \neg c)/e\}, \text{ and} \\ W & = & \{a, d, b \rightarrow \neg c\} \end{array}$$

This default theory has two Reiter extensions:
Firstly, $E = Cn(\{a, d, b, \neg c, e, b \to \neg c\})$ is a Reiter extension:

$$\begin{array}{rcl} E_0 & = & Cn(W), \\ E_1 & = & Cn(E_0 \cup \{b\}) = Cn(\{a, b, \neg c, d, b \to \neg c\}), \\ E_2 & = & Cn(E_1 \cup \{e\}) = Cn(\{a, b, \neg c, d, e, b \to \neg c\}), \\ E_3 & = & Cn(E_2 \cup \emptyset) = E_2, \\ E_i & = & E_2 \text{ for all } i > 3, \end{array}$$

and

$$E = \bigcup_{i=0}^{\infty} E_i = E_2.$$

A second Reiter extension is $F = Cn(\{a, d, c, \neg b, b \to \neg c\})$:

$$\begin{array}{rcl} F_0 & = & Cn(W), \\ F_1 & = & Cn(F_0 \cup \{c\}) = Cn(\{a, \neg b, c, d, b \to \neg c\}), \\ F_2 & = & Cn(F_1 \cup \emptyset) = F_1, \\ F_i & = & F_1 \text{ for all } i > 2, \end{array}$$

and

$$F = \bigcup_{i=0}^{\infty} F_i = F_1.$$

The temporal epistemic model $\mathcal{M}$ which corresponds with E, and the model $\mathcal{N}$ corresponding with F are shown in Figure 5.1.

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \\ \end{array} \quad \begin{array}{c} \boxed{\begin{array}{ccccc} 1 & 1 & 1 & 1 & \ldots \\ u & 1 & 1 & 1 & \ldots \\ u & 0 & 0 & 0 & \ldots \\ 1 & 1 & 1 & 1 & \ldots \\ u & u & 1 & 1 & \ldots \end{array}} \\ \mathcal{M} \end{array} \quad \begin{array}{c} \boxed{\begin{array}{ccccc} 1 & 1 & 1 & 1 & \ldots \\ u & 0 & 0 & 0 & \ldots \\ u & 1 & 1 & 1 & \ldots \\ 1 & 1 & 1 & 1 & \ldots \\ u & u & u & u & \ldots \end{array}} \\ \mathcal{N} \end{array}$$

Figure 5.1: Minimal models of $\tau(\Delta)$.

In this picture, time runs from left to right. Only the atoms are shown, where a 1 means the atom is known in $\mathcal{M}_t$, a 0 means that the negation of the atom is known, and a u means that neither the atom nor its negation is known (which means that $\mathcal{M}_t$ contains a valuation in which the atom is true and one in which it is false). Thus, for example, $\mathcal{M}_0$ contains all valuations m for which $m(a) = 1$ and $m(d) = 1$.

It is easy to verify that both $\mathcal{M}$ and $\mathcal{N}$ are conservative, and that $\tau(W)$ is true at all points. Furthermore, the temporal rules translating the default rules are:

$$\begin{array}{l} Ka \wedge \neg FK(\neg b) \rightarrow XKb \\ Kd \wedge \neg FK(\neg c) \rightarrow XKc \\ Kb \wedge \neg FKc \rightarrow XKe. \end{array}$$

Both models satisfy these rules. Moreover, both models minimally satisfy the requirements (with respect to the ordering $\preceq^g$ between temporal epistemic models). The correspondence between the Reiter extensions and the epistemic states can be described by

$$\begin{array}{ll} E_t = \mathrm{Th}(\mathcal{M}_t) & F_t = \mathrm{Th}(\mathcal{N}_t) \\ E = \mathrm{Th}(\lim \mathcal{M}) & F = \mathrm{Th}(\lim \mathcal{N}). \end{array}$$

In the following two Propositions 5.3 and 5.4 we will treat the two directions of the correspondence between Reiter extensions of a default theory and minimal temporal models of its temporal interpretation. Of course we can never hope to find a model of an inconsistent extension. Therefore we will assume that the set of axioms of a default theory is consistent, as this ensures that the extensions, if any exists, are consistent.

Proposition 5.3 Let $\Delta = \langle D, W \rangle$ be a default theory and $\mathcal{M}$ a $\preceq^g$-minimal temporal model of $\tau(\Delta)$. Then the set E defined by $E = Th(\lim \mathcal{M})$ is a Reiter extension of Δ. Moreover, $E_t = Th(\mathcal{M}_t)$ for all $t \in \mathbb{N}$.

Proof: Let $\mathcal{M}$ be a $\preceq^g$-minimal temporal model of $\tau(\Delta)$. First of all, observe that $\tau(\Delta)$ is a theory consisting of input and reasoning formulae satisfying the condition of Corollary 4.55. This means that $\mathcal{M}_0 = Mod(\{\gamma \mid K\gamma \in \tau(\Delta)\})$, and for each i, $\mathcal{M}_{i+1} = Mod(Th(\mathcal{M}_i) \cup \{\gamma \mid$ there is a rule $F_1 \wedge \ldots \wedge F_n \rightarrow XK\gamma \in \tau(\Delta)$ and $(\mathcal{M}, i) \models F_1 \wedge \ldots \wedge F_n\})$. Define $E = Th(\lim \mathcal{M})$. We will prove that $E_i = Th(\mathcal{M}_i)$ by induction on $i \in \mathbb{N}$, where the E_i are as defined in Definition 3.1.

- By definition, $E_0 = Cn(W)$ and $Cn(W) = Th(Mod(W)) = Th(Mod(\{\gamma \mid K\gamma \in \tau(\Delta)\}) = Th(\mathcal{M}_0)$.
- Now suppose $E_n = Th(\mathcal{M}_n)$. Then $E_{n+1} = Cn(E_n \cup \{\gamma \mid (\alpha : \beta_1, \ldots, \beta_k)/\gamma \in D, \alpha \in E_n$ and $\neg\beta_j \notin E$ for $1 \leq j \leq k\})$. For a rule $(\alpha : \beta_1, \ldots, \beta_k)/\gamma \in D$, we have $\alpha \in E_n$ if and only if $(\mathcal{M}, n) \models K\alpha$, and $\neg\beta_j \notin E$ iff $\lim \mathcal{M} \not\models K\neg\beta_j$. As $\mathcal{M}$ is closed by Proposition 4.46, this is equivalent to $(\mathcal{M}, n) \models \neg FK\neg\beta_j$. But this means that $E_{n+1} = Cn(Th(\mathcal{M}_n) \cup \{\gamma \mid$ there is a rule $F_1 \wedge \ldots \wedge F_n \rightarrow XK\gamma \in \tau(\Delta)$ and $(\mathcal{M}, i) \models F_1 \wedge \ldots \wedge F_n\}) = Th(Mod(Th(\mathcal{M}_n) \cup \{\gamma \mid$ there is a rule $F_1 \wedge \ldots \wedge F_n \rightarrow XK\gamma \in \tau(\Delta)$ and $(\mathcal{M}, i) \models F_1 \wedge \ldots \wedge F_n\}) = Th(\mathcal{M}_{n+1})$. But then it is easy to see that E is a Reiter extension of Δ, since $\bigcup_{i=0}^{\infty} E_i = \bigcup_{i=0}^{\infty} Th(\mathcal{M}_i) = Th(\lim \mathcal{M}) = E$. The second equality follows from the closedness of $\mathcal{M}$, see Proposition 4.9. $\square$

Proposition 5.4 Let $\Delta = \langle D, W \rangle$ be a default theory with W consistent and E a Reiter extension of Δ. Then the temporal epistemic model $\mathcal{M}$ defined by $\mathcal{M} = (Mod(E_t))_{t \in \mathbb{N}}$ is a $\preceq^g$-minimal temporal model of $\tau(\Delta)$ with $\lim \mathcal{M} = Mod(E)$.

Proof: Let E be a Reiter extension of Δ. Define $\mathcal{M} = (Mod(E_t))_{t \in \mathbb{N}}$. It is straightforward to check that $\lim \mathcal{M} = Mod(E)$, using the fact that $\bigcup_{i=0}^{\infty} E_i = E$, which follows as E is an extension. As in the proof of Proposition 5.3, it is the case that for any default $\delta = (\alpha, \beta_1, \ldots, \beta_k)/\gamma \in D$, it holds that $\alpha \in E_n$ and no $\beta_j \in E$ if and only if $\tau(\delta)$ is applicable in $\mathcal{M}$ at time point n. Using the characterization of Corollary 4.55, one can easily check that $\mathcal{M} \models_{\preceq^g} \tau(\Delta)$. □

We are now ready to state our main correspondence result between Reiter extensions and minimal temporal models (relying of course on Propositions 5.3 and 5.4), and the relation between their respective (sceptical) entailment relations.

Theorem 5.5 (Semantic correspondence) Let $\Delta = \langle D, W \rangle$ be a default theory with W consistent and let

$$\mathbb{M}(\tau(\Delta)) = \{\mathcal{M} \mid \mathcal{M} \text{ is a } \preceq^g \text{-minimal temporal model of } \tau(\Delta)\}.$$

1. By

$$\begin{array}{rcl} \Phi(E) & = & (Mod(E_t))_{t \in \mathbb{N}} \quad \text{and} \\ \Psi(\mathcal{M}) & = & Th(\lim \mathcal{M}) \end{array}$$

   two bijective mappings

$$\begin{array}{l} \Phi : \mathsf{Ext}(\Delta) \rightarrow \mathbb{M}(\tau(\Delta)) \\ \Psi : \mathbb{M}(\tau(\Delta)) \rightarrow \mathsf{Ext}(\Delta) \end{array}$$

   are defined that are each other's inverse. In other words, the equations

$$\begin{array}{rcl} \mathcal{M} & = & (Mod(E_t))_{t \in \mathbb{N}} \\ E & = & Th(\lim \mathcal{M}) \end{array}$$

   define a one-to-one correspondence between $E \in \mathsf{Ext}(\Delta)$ and $\mathcal{M} \in \mathbb{M}(\tau(\Delta))$.

2. For any propositional formula φ:

$$\varphi \text{ is a sceptical consequence of } \Delta \Leftrightarrow \tau(\Delta) \models_{\preceq^g} FK(\varphi).$$

This interpretation yields temporal semantics to default logic: given a default theory Δ, its semantics is given by $\mathbb{M}(\tau(\Delta))$. Note that no minimal models of $\tau(\Delta)$ exist if W is inconsistent (the default theory is classically inconsistent), or if Δ has no extensions (the default theory is nonmonotonically inconsistent). This is similar to other semantics for default logic.

A theory $\tau(\Delta)$ specifies default reasoning with the default theory Δ. It gives rise to the reasoning frame operator $\mathcal{T}_{\tau(\Delta)}$ as defined in Definition 4.63:

$$\mathcal{T}_{\tau(\Delta)}(X) = \{\mathcal{M} \mid \mathcal{M} \models_{\preceq^g} \tau(\Delta) \text{ and } Th(\mathcal{M}_0) = Cn(X)\}.$$

All minimal models $\mathcal{M}$ of $\tau(\Delta)$ actually have the same initial state $\mathcal{M}_0$, in which the agent believes the axioms of Δ, and nothing more. This means that $\mathcal{T}_{\tau(\Delta)}$ only assigns non-empty sets of traces to sets X whose propositional closure is equal to the set of axioms of Δ. So the above operator is not equal to the reasoning frame operator Tr_D of Chapter 3. By changing the notion of minimal model somewhat (essentially only allowing comparison of temporal models with the same initial state), the reasoning frame operator associated with the theory $\tau(D)$ is equal to Tr_D. This adapted notion of minimal model is introduced in Section 6.1 and is also used in Section 7.1.

In Theorem 5.5, we used $\preceq^g$-minimal consequence to derive properties of the form $FK(\varphi)$, the sceptical conclusions. But we can also infer other properties. For instance, we have that $\tau(\Delta) \models_{\preceq^g} FK\gamma \rightarrow FK\delta$ just in case δ belongs to every extension containing γ. Define $at_4 = PPPP\top \wedge HHHHH\bot$. Then we have that $\tau(\Delta) \models_{\preceq^g} at_4 \rightarrow K\gamma$ exactly when γ has been concluded at time point 4 in the trace of every extension. As a last example, $\tau(\Delta) \models_{\preceq^g} K\psi \rightarrow YK\varphi$ whenever φ is always derived before ψ.

We will now give some examples of monotonic reasoning about default logic. This reasoning will be done on TELC-models (see Definition 4.6). In Section 9.1.1 a proof system for validities on TELC-models is given.

Suppose we have the two default theories $\langle\{(\alpha : \beta)/\gamma\}, \{\alpha\}\rangle$ and $\langle\{(: \beta)/\gamma\}, \{\alpha\}\rangle$. Their temporal interpretations are $K\alpha \wedge (K\alpha \wedge \neg FK\neg\beta \rightarrow XK\gamma)$ and $K\alpha \wedge (\neg FK\neg\beta \rightarrow XK\gamma)$, which are equivalent on TELC-models, verifying that the two default theories are equivalent, in the sense that they have the same extensions. (In the proof system of Section 9.1.1, the equivalence of these two temporal formulae is easy to prove.)

Consider a default theory $\Delta = \langle D, W\rangle$ such that for some formulae φ, α, we have that $\varphi \in W$, and there is a default $(\varphi :)/\alpha \in D$. Then $\tau(\varphi) = K\varphi \in \tau(\Delta)$ and $\tau((\varphi :)/\alpha) = K\varphi \rightarrow XK\alpha \in \tau(\Delta)$. Since we have $K\varphi, K\varphi \rightarrow XK\alpha \models^c FK\alpha$, it immediately follows that $\tau(\Delta) \models_{\preceq^g} FK\alpha$, i.e. that α is a sceptical consequence of Δ.

As a last example, consider a normal default rule $(p : q)/q$. The temporal interpretation is $Kp \wedge \neg FK\neg q \rightarrow XKq$. By temporal reasoning we have that $Kp \wedge \neg FK\neg q \rightarrow XKq \models^c FKp \rightarrow (FKq \vee FK\neg q)$ which expresses the fact that if the prerequisite of the normal default rule is derived, then either the consequent or its negation will eventually be derived too (and this is independent of other default rules and axioms).

5.1.3 MTEL* and weak extensions

The proof of the correspondence between $\preceq^g$-minimal models and extensions is based on the characterization result of minimal models of special kinds of theories, notably Corollary 4.55. As this corollary also holds for MTEL*, the proof of the following proposition is completely analogous to the proofs of Propositions 5.3 and 5.4.

Proposition 5.6 (Semantic correspondence for MTEL*) Theorem 5.5 remains true when '$\preceq^g$-minimal' is replaced with '$\preceq^{gel}$-minimal', and '$\models_{\preceq^g}$' is replaced with '$\models_{\preceq^{gel}}$'.

In MTEL* we can also specify a variant of default logic, based on so-called *weak extensions* (introduced in [MT89]; see also [MT93]). In this variant, the reading of a default rule is changed, specifically the reading of the prerequisite. In order for a default rule $(\alpha : \beta)/\gamma$ to be applicable, it is no longer required that you believe α *now* (i.e., you have derived it using the axioms and other default rules), but it is sufficient that you believe α at *some future time* (i.e., α is believed in the limit). We start by giving the definition of a weak extension.

Definition 5.7 (Weak extension) Let $\Delta = \langle D, W\rangle$ be a default theory.

1. A set of sentences E is a *weak extension* of Δ if $E = \bigcup_{i=0}^{\infty} F_i$ where the sets F_i are defined as follows:

$$\begin{aligned} F_0 &= Cn(W), \text{ and for } i \geq 0: \\ F_{i+1} &= Cn(F_i \cup \{\gamma \mid (\alpha : \beta_1, \ldots, \beta_n)/\gamma \in D, \alpha \in E \\ &\qquad \text{and } \neg\beta_j \notin E \text{ for } 1 \leq j \leq n\}). \end{aligned}$$

2. The set of weak extensions of Δ is denoted by $\mathsf{Ext}_w(D, W)$.
3. If a formula φ occurs in all weak extensions of Δ, it is called a *weak sceptical consequence* of Δ.

The new reading of (the prerequisite of) a default rule suggests a new translation.

Definition 5.8 Let $\Delta = \langle D, W\rangle$ be a default theory.

1. The mapping σ from default rules to TEL-formulae is defined by

$$\sigma : (\alpha : \beta_1, \ldots, \beta_n)/\gamma \mapsto FK\alpha \wedge \neg FK(\neg\beta_1) \wedge \ldots \wedge \neg FK(\neg\beta_n) \rightarrow XK\gamma.$$

2. The mapping σ is extended to default theories by

$$\begin{aligned} &\sigma(D) = \{\sigma(\delta) \mid \delta \in D\}, \\ &\sigma(W) = \tau(W), \text{ and} \\ &\sigma(\Delta) = \sigma(D) \cup \sigma(W). \end{aligned}$$

This translation specifies default reasoning with weak extensions.

Proposition 5.9 Let $\Delta = \langle D, W \rangle$ be a default theory with W consistent.

1. Suppose $\mathcal{M}$ is a $\preceq^{gel}$-minimal model of $\sigma(\Delta)$. Then the set $E = Th(\lim \mathcal{M})$ is a weak extension of Δ, and $F_i = Th(\mathcal{M}_i)$ for all $i \in \mathbb{N}$ (where the sets F_i are as in Definition 5.7).
2. Suppose E is a weak extension of Δ. Then the temporal model $\mathcal{M}$ defined by $\mathcal{M} = (Mod(F_t))_{t \in \mathbb{N}}$ is a $\preceq^{gel}$-minimal model of $\sigma(\Delta)$.
3. For any propositional formula φ:

$$\varphi \text{ is a weak sceptical consequence of } \Delta \Leftrightarrow \sigma(\Delta) \models_{\preceq^{gel}} FK\varphi.$$

Proof: This proof is essentially analogous to the proofs of Propositions 5.3 and 5.4, using the fact that Corollary 4.55 also holds for MTEL*, and the fact that for closed models, it holds that $(\mathcal{M}, i) \models FK\alpha$ if and only if $\lim \mathcal{M} \models_{S5} K\alpha$. □

In MTEL*, nothing prevents us from mixing the two kinds of reading of the prerequisites. One can consider rules of the form

$$K\alpha \wedge FK\beta \wedge \neg FK(\neg\gamma_1) \wedge \ldots \wedge \neg FK(\neg\gamma_k) \rightarrow XK\omega$$

in which a prerequisite α is interpreted according to the traditional reading, and a prerequisite β which has the weak interpretation. Instead of interpreting all prerequisites in a default theory in the same way, one can distinguish between weak and strong prerequisites. This suggests a new, more flexible, variant of default logic we briefly describe.

Definition 5.10 (Weak/strong default logic)

1. A WS-default theory is a pair $\langle D, W \rangle$ where W is a set of propositional formulae, and D is a set of expressions (called *WS-default rules*) of the form

$$(\alpha, \beta : \gamma_1, \ldots, \gamma_k)/\omega$$

   where α, β, γ_1 through γ_k, and ω are propositional formulae.

2. A set of propositional formulae E is a *WS-extension* of $\langle D, W \rangle$ if $E = \bigcup_{i=0}^{\infty} G_i$ where the sets G_i are defined as follows:

$$\begin{aligned} G_0 &= Cn(W), \text{ and for } i \geq 0: \\ G_{i+1} &= Cn(G_i \cup \{\omega \mid (\alpha, \beta : \gamma_1, \ldots, \gamma_n)/\omega \in D, \alpha \in G_i, \beta \in E, \\ &\qquad \text{and } \neg\gamma_j \notin E \text{ for } 1 \leq j \leq n\}). \end{aligned}$$

In the literature a number of approaches to giving semantics to default logic exist. In the next section we will compare them with our temporal approach.

5.1.4 Other approaches to semantics for default logic

In this section we will compare our approach with other approaches to semantics for default logic, as known from literature: [Gab82], [Eth87], [BS94], [MST93], [ACP97], and [LS92].

Comparison with Gabbay

In [Gab82] an approach to nonmonotonic logic is described where intuitionistic logic is used as a basis. The semantics are described by Kripke models (in the form of temporal frames) where the accessibility relation is a pre-ordering on the worlds according to (time) points that describe the stages in the reasoning (a well-known approach to the semantics of intuitionistic logic; see [Kri65]). This idea of using temporal frames to represent the flow of time of the reasoning process itself is in common with our approach. However, there are differences as well. As Gabbay's approach does not use epistemic states, one always has to commit to justifications: it is not possible to express that the truth value of a justification β should be left open in the future of the reasoning process. In our approach there is a choice: either one can choose to commit to justifications or not. The second case is described in the current section, while a slight modification of the temporal translation of the default rules will enable our approach to commit to justifications. This will not be worked out here. A second difference with Gabbay's approach is that we do not give temporal interpretations to classical connectives such as negation and implication, whereas the intuitionistic approach does: e.g., $\neg\alpha$ is true at a time point if and only if for all future time points α is false. A third difference is that in our case there is a time difference (in principle of one step) between the conclusion γ of a default rule $(\alpha : \beta)/\gamma$ and its condition α. In Gabbay's approach both α and γ refer to the same time point, while only β is interpreted in a temporal manner. We interpret both β and γ in a temporal manner. This essentially means that in Gabbay's approach default reasoning steps are not counted by the time measure as used. This difference has rather far-reaching implications for the models. In Gabbay's case the conclusions of the reasoning process are meant as those statements that are true at all time points of the intended model, whereas in our case they are the statements that are ('become') known to be true at some time point of the model, i.e., that are known to be true in the limit model. Gabbay's logic does not yield semantics for Reiter's default logic. In [Łuk90], pp. 149-154, a critical analysis is given of Gabbay's approach.

Comparison with Etherington

In [Eth87] it is argued that a semantics of default logic in terms of typical semantic structures as known is not possible, because the outcome of a default reasoning process essentially depends on *the way knowledge is extended* (see page 497 of [Eth87]), and this requires knowledge that is not inherent in typical semantical structures. Precisely this view was our motivation to model the traces of the reasoning process explicitly in our semantics. Etherington's semantics has some similarities and some differences with our approach. The main similarity is that our minimal temporal models correspond to (maximal) chains in the sense of Etherington's preference ordering. A maximal element with respect to Etherington's ordering corresponds to our notion of limit model. Actually we define the (temporal) ordering relation between states in the reasoning process in a logical manner by temporal axioms (the temporal translations of the default rules), whereas Etherington gives a more ad hoc definition of his preference relation. Our notion of minimality with respect to the usual refinement relation corresponds to what in Etherington's case is also hidden in the definition of the preference relation, namely that nothing else can happen than what is based on (generated by) the given defaults (a kind of groundedness-condition).

Comparison with Besnard and Schaub

The approach of Besnard and Schaub [BS94] is similar to an earlier approach to semantics for Reiter's default logic by Schaub [Sch91]. Instead of pairs of classes of interpretations, one for the formulae in an extension, one for the justifications, Besnard and Schaub use classes of Kripke models with one *actual world*, where the formulae of an extension have to be true in the actual worlds, and the worlds reachable from the actual world are used for the justifications. Also an ordering $<_{D'}$ is defined on classes of Kripke models, which depends on the defaults in the default theory. Although both in their approach and ours, Kripke models are used, the way in which they are used is quite different, not to mention the fact that we use epistemic states, and Besnard and Schaub use two-valued models for the extensions. As in Etherington's approach, maximal chains in their ordering correspond to our minimal temporal models, and Besnard and Schaub also give a more ad hoc definition of their precedence relation. As we want our models to reflect the reasoning path which leads to an extension, it was natural to use a linear time model, and as at any point in time, not all facts will be known when reasoning, the use of epistemic states seems justified. Both their approach and ours use minimization of models with respect to a preference relation. Their ordering $<_{D'}$ depends on the default theory, whereas our ordering $\preceq^g$ is structurally defined, independent of the defaults.

Comparison with Marek, Schwarz and Truszczyński

There is a long tradition of research into modal nonmonotonic logics starting with [MD80]. With every modal logic of knowledge (of belief) one can associate a nonmonotonic logic based on it. Given a theory I in the modal language, an expansion T is a theory in this language satisfying a certain fixpoint definition. These expansions play a role similar to the role extensions play in default logic. For a number of modal logics (most notably a logic called S4F), it is possible to translate a default theory Δ into a theory I_Δ such that expansions of I_Δ correspond to extensions of Δ. The translation of a default rule is quite similar to our definition: a rule $(\alpha : \beta)/\gamma$ is translated into the modal formula $L\alpha \wedge LM\beta \rightarrow \gamma$, where $M = \neg L \neg$ by definition

([MST93], see also [MT89, MT92, ST94, Sch95b]). This rule is to be read as: if you know α and you know β is possible, then γ is true. The essential difference with our approach is that the modal rule of [MST93] can be seen as a static (closure) condition on the beliefs of the agent. Any set of beliefs that can be regarded as the set of beliefs of a rational introspective agent (that is, it must be an expansion), must be closed under the default rules. Our translation, in contrast, emphasizes the dynamic (behavioral) aspect. A minimal temporal epistemic model does not describe a belief set of an agent, but a reasoning trace of a rational introspective agent. The limit of such a model corresponds to a (final) belief set. Thus, our logic does not fall into the general framework of Marek, Schwarz, and Truszczyński. Our underlying (monotonic) logic, temporal epistemic logic, is essentially just standard S5, with a straightforward temporalization over the natural numbers. So we use simpler techniques than the approach of [MST93] which is based on S4F (or a modal logic in-between the logics N and S4F. In [ACGP96] it is shown that, with a slight adaptation of the fixpoint equation and the translation, logics between KD4 and KD4Z can also be used).

Comparison with Amati, Aiello and Pirri

The idea of Marek, Schwarz and Truszczyński is taken even further in [ACP97], where it is shown that extensions correspond to certain theorems in the modal logic KD4Z. The fixpoint for extensions is expressible in the language. Thus, the fixpoint is not a construction on top of the logic, but extensions correspond to fixpoints expressed in the language, provable in KD4Z. Again, however, this is a static description of the set of beliefs of an agent. Our perspective is different: we want to make the construction an explicit temporal process as performed by the agent.

Comparison with Lin and Shoham

In [LS92] a bimodal logic of knowledge and assumptions is described. Since this logic is treated in more detail in Section 5.5, we will not say too much about it here. Suffice it to say that the major difference between our approach and theirs is again in the perspective: their perspective is static, with a characterization result that implicitly gives a dynamic description. In our approach, the dynamic perspective is the most important, and the static notion of limit is derived from the dynamic notions.

5.1.5 Concluding remarks

In this section we have given a temporal interpretation to the notion of a justification in a default rule. This enables one to use concepts from temporal logic to study default reasoning. Of course such a translation does not automatically imply that the problems of default logic will be solved at once. Temporal epistemic logic with minimality has its own complexity.

The interpretation and correspondence yield a temporal semantics for default logic. Although other approaches in the literature for giving semantics to default logic exist, we feel that making the temporal aspect explicit in a formalism where the dynamics of the reasoning process (choosing the default assumptions) have an impact on the final outcome, gives a clear and intuitively appealing meaning to default logic. A similar underlying idea was later also used in an approach of Lin and Reiter to give semantics for logic programming in the Situation Calculus ([LR96]; see also Section 5.8).

In [HMT95] temporalized epistemic default logic (TEDL) is introduced. A similarity with the approach introduced here is that a dynamic perspective on default reasoning is used. However, there are two differences. The first difference is that TEDL formalizes a default logic quite different from Reiter's default logic: in TEDL the justifications refer only to the current state of knowledge; knowledge that is acquired at later points in time is not taken into account. The notion of extension, or final conclusion set, is defined in a constructive manner; no fix point definition is used. This implies that the conclusion sets are different from Reiter extensions: a TEDL-conclusion set E may be based on (generated by) consequents of default rules for which the prerequisite is included in E, but the negation of the justification is also in E. A second difference is that the semantics of TEDL is defined using labeled branching time temporal models.

5.2 Default logic: the branching time case

In the previous section, we gave a translation of default theories into linear time logic. Using the ideas and techniques from Section 4.3, we will give a specification of default logic in a branching time variant of MTEL. We will see that (under a particular topological condition, called extension completeness) one branching time model can be constructed in which precisely all possible lines of reasoning (and the resulting conclusion sets) can be represented (even though they might be mutually contradictory). The semantics of the default theory can be defined on the basis of this single model. In particular, we show how sceptical and credulous entailment relations can be defined on the basis of this model.

5.2.1 Minimal branching time epistemic logic

The branching time temporal logic we will use, is essentially the branching time variant of MTEL. In Section 4.3, a branching time logic was defined, parameterized by the choice of the underlying information state frame. The information state frame we choose here is IS^{ep} (see Definition 2.5) restricted to closed (Definitiondf:closedmodel) models. The C-operator used in Section 4.3, is replaced by a K-operator; the semantics of the C-operator when using IS^{ep} coincides with the semantics of the S5-operator K when we view an information state from IS^{ep} as an S5-model. The resulting logic, branching time epistemic logic, can be seen as the result of temporalizing S5 using branching time models, just as TEL is the result of temporalizing S5 using the natural numbers as flow of time.

In the specification of default logic in temporal logic, we used an ordering on temporal models. When extending that approach to the branching time case, we also need to extend the ordering $\preceq^g$ of MTEL to branching time models. The basic idea is the same: we compare the states pointwise, using the ordering $\preceq$ of the information state frame. To be able to do a pointwise comparison of two models, they have to be based on the same flow of time.

Definition 5.11 The ordering $\preceq^g_{br}$ between branching time temporal models is defined by: $\mathcal{M} \preceq^g_{br} \mathcal{N}$ if they have the same flow of time and $\mathcal{M}(t) \preceq \mathcal{N}(t)$ for all time points t.

This ordering can be used to define minimal models and minimal entailment, analogously to Section 4.4.

Definition 5.12 For a set K of temporal formulae, we say that a temporal model $\mathcal{M}$ is a $\preceq^g_{br}$-*minimal* model of K if $\mathcal{M} \models K$ and whenever a model $\mathcal{N} \preceq^g_{br} \mathcal{M}$ is a model of K then $\mathcal{N} = \mathcal{M}$. If $\mathfrak{T}$ is a temporal theory, then by $\mathsf{MLT}(\mathfrak{T})$ we denote the set of all $\preceq^g_{br}$-minimal *linear time* models of $\mathfrak{T}$.

The branches in a branching time model represent traces, and under certain circumstances these traces may have a limit, so that we can talk about the limit of a branch in a model. These limits exist if the traces are conservative.

Definition 5.13 (Limits in a conservative model) Let $\mathcal{M}$ be a temporal model.

1. $\mathcal{M}$ is *conservative* if $\mathcal{M}(t) \preceq \mathcal{M}(s)$ whenever $t \ll s$.
2. The *limit* of a branch $\mathcal{B}$ in a model $\mathcal{M}$, denoted $\lim_{\mathcal{B}}\mathcal{M}$, is the limit of $\mathcal{B}$ viewed as a trace in IS^{ep} (according to Definition 2.20). If $\mathcal{B} = \mathcal{M}$, we will simply write $\lim \mathcal{M}$.

5.2.2 Interpreting default logic in branching time temporal logic

In Section 5.1 a translation of default theories into temporal theories of linear time temporal epistemic logic was given. The translation into branching time temporal epistemic logic uses the same temporal interpretation of a default rule $(\alpha : \beta)/\gamma$:

> *If the agent knows α now, and β remains consistent in the future, then it may conclude γ at the next point in time.*

The question is how we should interpret consistency in the future, and what the next point in time is: should β be consistent in *all* or in *some* possible futures (branches), and should γ be concluded in all or in some next points? As each branch should represent a valid default reasoning path, we want the above default applicability rule to be satisfied by every branch in a model. This means that if the agent knows α at some point in time, and there is a future (a branch) starting at this point along which β remains consistent, then on the next point in time on this branch, γ must be concluded. We can not express this *directly* in our branching time logic, but we can do it by describing the rule one point later: if the agent knows α at the *previous* point in time, and there is a branch along which $\neg\beta$ is not known, then we have to conclude γ *now*. In conservative models (and we will require models to be conservative), α is known at the previous point in time if and only if it is known sometimes in the past. Furthermore, $\neg\beta$ is not known along some branch precisely if it is not the case that $\neg\beta$ is known sometimes in the future along all branches. The translation of a default rule $(\alpha : \beta)/\gamma$ then becomes

$$P(K\alpha) \wedge \neg\forall \mathrm{F}(K\neg\beta) \rightarrow K\gamma.$$

In case there are more justifications, $\beta_1, \ldots, \beta_n$, the translation is $P(K\alpha) \wedge \neg\forall \mathrm{F}(K\neg\beta_1 \vee \ldots \vee K\neg\beta_n) \rightarrow K\gamma$, but for clarity we will consider only defaults with one justification in the remainder of this section. Instead of reducing the class of models to the class of conservative models, we will add formulae $P(K\alpha) \rightarrow K\alpha$ (for α propositional) that ensure conservativity. As was the case in the previous section, the axioms of the default theory can just be prefixed by a K-operator.

As we do not want any extra conclusions in the corresponding model than those which have to be drawn, we will take the minimal models with respect to $\preceq^g_{br}$.

Definition 5.14 (Branching time interpretation of a default theory) Let $\Delta = \langle D, W\rangle$ be a default theory. Define

$$\begin{array}{lcl} \mathit{Cons} & = & \{P(K\alpha) \rightarrow K\alpha \mid \alpha \text{ a propositional formula}\}, \\ D' & = & \{P(K\alpha) \wedge \neg\forall \mathrm{F}(K\neg\beta) \rightarrow K\gamma \mid (\alpha : \beta)/\gamma \in D\}, \text{ and} \\ W' & = & \{K\alpha \mid \alpha \in W\}. \end{array}$$

The temporal interpretation of Δ is the temporal theory $T_\Delta = \mathit{Cons} \cup D' \cup W'$. The set of minimal linear time models of T_Δ is denoted by $\mathsf{MLT}(\Delta)$.

On linear time models, T_Δ does the same as $\tau(\Delta)$ (see Definition 5.1).

Theorem 5.15 Let $\Delta = \langle D, W\rangle$ be a default theory.

1. If $\mathcal{M}$ is a minimal linear time temporal model of T_Δ, then $Th(\lim \mathcal{M})$ is a Reiter extension E of Δ. Moreover, $E_i = Th(\mathcal{M}_i)$ for all $i \in \mathbb{N}$.

2. If W is consistent and E is a Reiter extension of Δ, then the temporal model $\mathcal{M}$ defined by $\mathcal{M} = (\mathit{Mod}(E_i))_{i\in\mathbb{N}}$ is a minimal linear time temporal model of T_Δ with $Th(\lim \mathcal{M}) = E$.

Proof: This result is essentially the same as Propositions 5.3 and 5.4. We will only make some remarks about the different translation. It is easy to see that on linear models, $\neg F(K\neg\beta)$ is equivalent to $\neg\forall\mathrm{F}(K\neg\beta)$. We will show that any conservative linear model satisfies $K\alpha \wedge \neg F(K\neg\beta) \rightarrow XK\gamma$ if and only if it satisfies $P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta) \rightarrow K\gamma$. Suppose that $\mathcal{M} \models K\alpha \wedge \neg F(K\neg\beta) \rightarrow XK\gamma$ and that for some $t \in \mathbb{N}$, $(\mathcal{M}, t) \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$. Remark that $(\mathcal{M}, t) \models P(K\alpha)$ implies that $t > 0$. Then it easily follows (given conservativity) that $(\mathcal{M}, t-1) \models K\alpha \wedge \neg F(K\neg\beta)$, so $(\mathcal{M}, t-1) \models XK\gamma$ whence $(\mathcal{M}, t) \models K\gamma$. Now suppose that $\mathcal{M} \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta) \rightarrow K\gamma$ and for some $t \in \mathbb{N}$, $(\mathcal{M}, t) \models K\alpha \wedge \neg F(K\neg\beta)$. Then $(\mathcal{M}, t+1) \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$, so $(\mathcal{M}, t+1) \models K\gamma$. It follows that $(\mathcal{M}, t) \models XK\gamma$. $\square$

For the case of lines of default reasoning that do not stabilize after a finite number of steps, topological properties of the space of reasoning patterns become relevant. Before defining a metric on the space of linear time models, we recall the following well-known topological definitions. A sequence $(a_i)_{i\in\mathbb{N}}$ in a metric space X with metric d is called *convergent* with limit $a \in X$ if for each $\varepsilon > 0$ there exists an $N \in \mathbb{N}$ such that for all $i \geq N$ it holds that $d(a_i, a) < \varepsilon$. A subset Y of X is called *closed* if for every convergent sequence in X with all a_i in Y, its limit is included in Y.

Definition 5.16 (Metric) Define the following metric d on the set of linear time models LT: for $\mathcal{M}, \mathcal{N}$ linear models:

$$d(\mathcal{M}, \mathcal{N}) = \begin{cases} 0 & \text{if } \mathcal{M} = \mathcal{N} \\ 2^{-i} & \text{where } i = \sup\{j \in \mathbb{N} \mid \forall k \leq j : \mathcal{M}_k = \mathcal{N}_k\}, \text{ otherwise.} \end{cases}$$

It is easy to see that the metric space (LT, d) is complete, i.e., that every Cauchy-sequence has a limit. The following definition will play an important role in the next subsection:

Definition 5.17 (Extension complete) A default theory Δ is called *extension complete* if $\mathsf{MLT}(\Delta)$ is a closed subset of the metric space (LT, d).

Proposition 5.18 Every default theory with a finite set of defaults is extension complete.

Proof: A default theory Δ with a finite set of defaults has finitely many extensions (this follows easily from the fact that every extension is the propositional closure of W and the set of generating defaults, see [Rei80b]), so by Theorem 5.15 $\mathsf{MLT}(\Delta)$ is finite. In a metric space, all finite sets are closed. $\square$

As an example of a default theory which is *not* extension complete, let $\Delta = \langle D, W \rangle$, with

$$\begin{array}{lcl} D & = & \{: b/b\} \cup \{(a_i : a_{i+1})/a_{i+1} \mid i \in \mathbb{N}\} \cup \{(a_i : \neg a_{i+1})/\neg a_{i+1} \mid i \in \mathbb{N}\} \text{ and} \\ W & = & \{a_0\} \cup \{b \to a_i \mid i \in \mathbb{N}\}. \end{array}$$

This (normal) default theory has infinitely many extensions: $F = Cn(W \cup \{b\})$ and for each $n \in \mathbb{N}$, $E^{(n)} = Cn(W \cup \{a_i \mid i \leq n\} \cup \{\neg a_{n+1}\})$. In this example, the linear time models corresponding to these extensions form a convergent sequence in (LT, d), but its limit is the model $(\mathcal{M}_t)_{t\in\mathbb{N}}$, with $\mathcal{M}_t = Mod(W \cup \{a_i \mid i \in \mathbb{N}\})$, which is not in $\mathsf{MLT}(\Delta)$ (see Figure 5.2). In Figure 5.2, we have indicated the (non-trivial) formulae that are true in the various time points, where a formula is not repeated if it was true earlier.

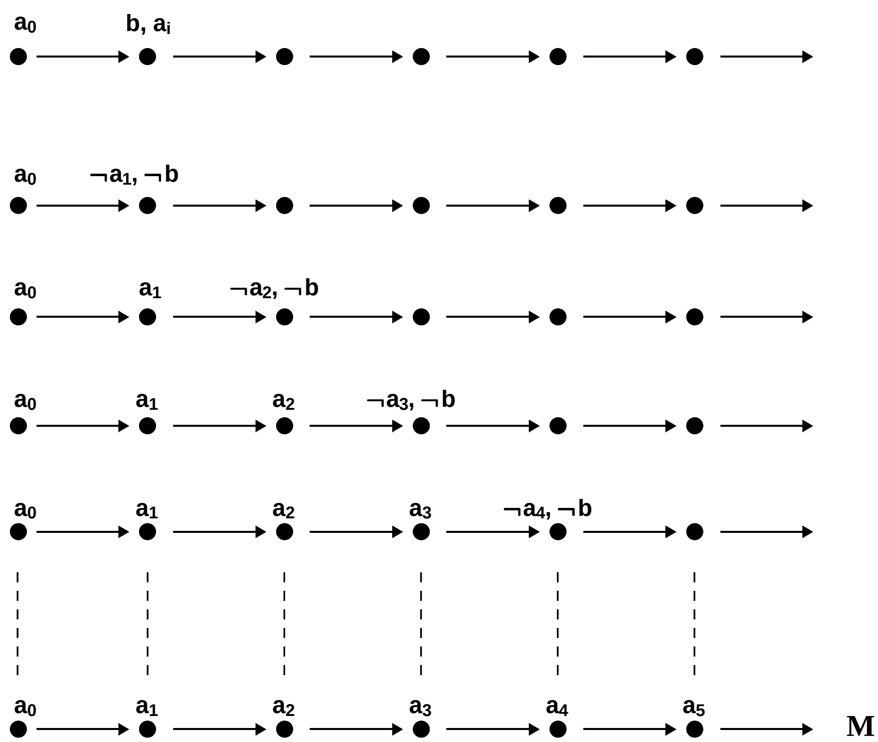

Figure 5.2: Traces of a theory which is not extension complete.

5.2.3 Joint embeddings of linear time models of default theories

The previous section provided semantics for default reasoning in the form of a set of linear time models that represent the possible default reasoning patterns. An alternative manner of representing these reasoning patterns is by means of *one* branching time model, where each branch represents one alternative reasoning pattern (with a Reiter extension as its limit). This provides semantics for default reasoning in the form of one "standard" model. The aim of this section is for any given default theory to indeed construct such a branching time model, under certain conditions (viz. extension completeness). To this end we apply the constructions of Section 4.3 to the model theory of the temporal translation of default theories.

Section 4.3 gave some results on preservation of modelhood by the constructions of closure, coproduct and joint closure. In addition to those, we need results about minimal modelhood. In the sequel, let $*C$ denote the coproduct of a set of models C (see Definition 4.30). The coproduct construction preserves minimal models.

Proposition 5.19 Let K be a temporal theory and let $\mathfrak{B}$ be a set of models. Then all models in $\mathfrak{B}$ are minimal models of K if and only if $*\mathfrak{B}$ is a minimal model of K.

Proof: Suppose $\mathfrak{B}$ is a set of minimal models of a temporal theory K, and let $*\mathfrak{B}$ be its coproduct. Since the evaluation of a formula in a point depends only on the connected component in which it lies, it is easy to see that $*\mathfrak{B}$ is a model of K. Now suppose that there is a smaller model $\mathcal{M}$ of K. Then there is a point $s \in \mathcal{M}$ such that $\mathcal{M}_s \prec (*\mathfrak{B})_s$. This point s is an element of one of the models $\mathcal{N}$ in $\mathfrak{B}$. Now let us look at the model $\mathcal{M}'$ which is the restriction of $\mathcal{M}$ to the flow of time of $\mathcal{N}$. It is easy to verify that $\mathcal{M}' \prec^g_{br} \mathcal{N}$ and that $\mathcal{M}'$ is a model of K, contradicting the assumption that $\mathfrak{B}$ contains only minimal models of K. Thus, $*\mathfrak{B}$ is a minimal model of K.

For the other direction, suppose $\mathfrak{B}$ contains a model $\mathcal{M}$ which is not a minimal model of K. If it is not a model of K, then it is easy to see that $*\mathfrak{B}$ can not be a model of K. Otherwise there is a model $\mathcal{N} \prec^g_{br} \mathcal{M}$ which is a model of K. Consider $\mathfrak{B}' = (\mathfrak{B} \setminus \{\mathcal{M}\}) \cup \{\mathcal{N}\}$. Then $*\mathfrak{B}' \prec^g_{br} *\mathfrak{B}$ and $*\mathfrak{B}'$ is a model of K, so $*\mathfrak{B}$ is not a minimal model of K. □

As we want to study minimal models of T_Δ and connections between them, the following proposition is useful:

Proposition 5.20 Let Δ be a default theory, and let $f : \mathcal{M} \to \mathcal{M}'$ be a homomorphism such that for every branch $\mathcal{B}'$ in $\mathcal{M}'$, there is a branch $\mathcal{B}$ in $\mathcal{M}$ such that $f[\mathcal{B}] = \mathcal{B}'$. If $\mathcal{M}$ is a minimal model of T_Δ then $\mathcal{M}'$ is also a minimal model of T_Δ.

Proof: Suppose $\mathcal{M}$ has flow of time $(T, <)$ and $\mathcal{M}'$ has flow of time $(T', <')$. First we will show that $\mathcal{M}'$ is a model of T_Δ. Unfortunately, we can not use Proposition 4.29, since f is not necessarily branch-surjective. Take a point $s' \in T'$. Then s' lies on at least one branch, say $\mathcal{B}'$. Given the requirement on f, there must be a branch $\mathcal{B}$ in $\mathcal{M}$ such that $f[\mathcal{B}] = \mathcal{B}'$. Note that $\mathcal{B}'$ is an isomorphic copy of $\mathcal{B}$. It follows that $(\mathcal{M}', s') \models W'$ and $(\mathcal{M}', s') \models \mathit{Cons}$ (remember that $\mathit{Cons} \subseteq T_\Delta$; see Definition 5.14).

Now take a rule $P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta) \rightarrow K\gamma$ in D', and suppose $(\mathcal{M}', s') \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$. This means that there must exist a branch $\mathcal{B}'$ in $\mathcal{M}'$ such that s' lies on $\mathcal{B}'$, there is a $t' \in \mathcal{B}'$ with $t' \ll s'$ and $\mathcal{M}'(t') \models K\alpha$, and for all $u' \in \mathcal{B}'$: if $s' \ll u'$ then $\mathcal{M}'(u') \not\models K\neg\beta$. Given the requirement on f, there is a branch $\mathcal{B}$ in $\mathcal{M}$ with $f[\mathcal{B}] = \mathcal{B}'$. Thus, there is a (unique) $s \in \mathcal{B}$ with $f(s) = s'$, and it is easy to verify that $(\mathcal{M}, s) \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$. But then $(\mathcal{M}, s) \models K\gamma$, as $\mathcal{M}$ is a model of T_Δ, and therefore $(\mathcal{M}', s') \models K\gamma$. We have proved that $\mathcal{M}'$ is a model of T_Δ.

Suppose that $\mathcal{M}'$ is not minimal, then there exists a model $\mathcal{N}' \prec^g_{br} \mathcal{M}'$, such that $\mathcal{N}' \models T_\Delta$. We will define a model $\mathcal{N}$ of T_Δ which is smaller than $\mathcal{M}$, contradicting the hypothesis that $\mathcal{M}$ is minimal. Let $\mathcal{N}$ be based on the flow of time $(T, <)$, and define $\mathcal{N}(s) = \mathcal{N}'(f(s))$. Then $\mathcal{N}(s) = \mathcal{N}'(f(s)) \preceq \mathcal{M}'(f(s)) = \mathcal{M}(s)$ (the definition of homomorphism in fact only requires that $\mathcal{M}'(f(s)) \equiv \mathcal{M}(s)$, but for closed models equivalence implies equality), and there is at least one point $u' \in T'$ such that $\mathcal{N}'(u') \neq \mathcal{M}'(u')$. But as f is certainly surjective, there is a $u \in T$ with $f(u) = u'$, so we have that $\mathcal{N}(u) \neq \mathcal{M}(u)$. Take a point $s \in T$, then the path from the root of the tree in which s lies is mapped isomorphically to the path from a root to $f(s)$, so since $\mathcal{N}'$ is a model of Cons and W', it is easy to see that $(\mathcal{N}, s) \models \mathit{Cons} \cup W'$. Now take a rule $P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta) \rightarrow K\gamma$ in D' and suppose $(\mathcal{N}, s) \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$. This means that there is a branch $\mathcal{B}$ in $\mathcal{N}$ on which s lies, such that there is a $t \ll s$ with $\mathcal{N}(t) \models K\alpha$ and for all $u \in \mathcal{B}$ with $s \ll u$, $\mathcal{N}(u) \not\models K\neg\beta$. But then $f[\mathcal{B}]$ is a branch in $\mathcal{N}'$ with $f(t) \ll f(s)$ and $\mathcal{N}'(f(t)) \models K\alpha$, and for all $u' \in \mathcal{B}'$ with $s \ll u'$ it must be the case that $u' = f(u)$ for some $u \in \mathcal{B}$ with $s \ll u$, so $\mathcal{N}(u) = \mathcal{N}'(f(u)) \not\models \neg K\beta$. As $\mathcal{N}'$ is a model of D', we have $\mathcal{N}'(f(s)) \models K\gamma$, so that $\mathcal{N}(s) \models K\gamma$. Thus $\mathcal{N}$ is a model of D', so it is a model of T_Δ, in contradiction with the hypothesis that $\mathcal{M}$ was a minimal model of T_Δ. We have proved that $\mathcal{M}'$ is a minimal model of T_Δ. □

Sometimes properties of branching time temporal models can be related to properties of the linear time models that are their branches. In our case we have the following results for the property of being a minimal model of T_Δ.

Theorem 5.21 Let Δ be a default theory.

1. If $\mathcal{M}$ is a (branching time) temporal model such that $\mathsf{Br}(\mathcal{M}) \subseteq \mathsf{MLT}(\Delta)$, then $\mathcal{M}$ is a minimal model of T_Δ.

2. Suppose Δ is extension complete and $\mathfrak{B} \subseteq \mathsf{MLT}(\Delta)$. If $f : *\mathfrak{B} \rightarrow \mathcal{M}$ is a surjective homomorphism, then $\mathcal{M}$ is a minimal model of T_Δ.

Proof:

1. Suppose $\mathsf{Br}(\mathcal{M}) \subseteq \mathsf{MLT}(\Delta)$. By definition, $\mathsf{MLT}(\Delta)$ consists of (linear) minimal models of T_Δ, so the same holds for $\mathsf{Br}(\mathcal{M})$. By Proposition 5.19, the coproduct $*\mathsf{Br}(\mathcal{M})$ is a minimal model of T_Δ. Now define the function $f : *\mathsf{Br}(\mathcal{M}) \to \mathcal{M}$ mapping every branch in $*\mathsf{Br}(\mathcal{M})$ into $\mathcal{M}$. It is easy to see that f satisfies the requirement of Proposition 5.20, so $\mathcal{M}$ is a minimal model of T_Δ.

2. We will show that $\mathsf{Br}(\mathcal{M}) \subseteq \mathsf{MLT}(\Delta)$, from which the desired result follows by part 1. Take any branch $\mathcal{D}$ of $\mathcal{M}$, and assume (without loss of generality) that it has the natural numbers as flow of time. Now take an arbitrary $n \in \mathbb{N}$. Since f is surjective, there must be a point $s \in *\mathfrak{B}$ such that $f(s) = n$. This point s must lie on a branch $\mathcal{D}'$ of $*\mathfrak{B}$, and this $\mathcal{D}'$ is a linear time model in $\mathfrak{B}$. From the definition of homomorphism, it follows that f maps this branch up to point s isomorphically onto $\mathcal{D}$ (up to point n). This means that $d(\mathcal{D}, \mathcal{D}') \leq 2^{-n}$. As n was chosen arbitrarily, we can find a sequence of linear time models in $\mathfrak{B}$ that have $\mathcal{D}$ as their limit. The models of $\mathfrak{B}$ are in $\mathsf{MLT}(\Delta)$, which is closed as Δ is extension complete. This means that $\mathcal{D} \in \mathsf{MLT}(\Delta)$. By part 1 we have that $\mathcal{M}$ is a minimal model of T_Δ.

$\square$

It can easily be shown that in general minimal models of T_Δ can have branches that are *not* in $\mathsf{MLT}(\Delta)$. Therefore, the class of all minimal branching time models of T_Δ cannot be considered a suitable semantics for default logic (this is different, however, for normal default theories; see Subsection 5.2.5). Consider the default theory $\Delta = \langle D, W \rangle$ with $D = \{(\top : a)/a, (\top : c)/b, (a : c)/c, (a : \neg c)/\neg c\}$ and $W = \emptyset$, and the model $\mathcal{M}$ of Figure 5.3. In Figure 5.3 we have again indicated the

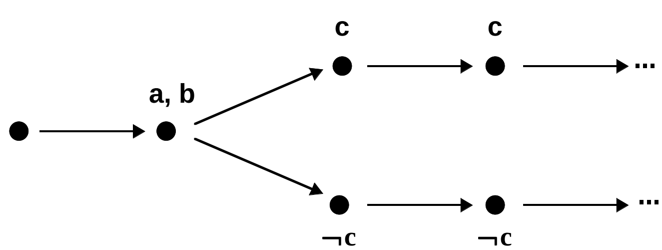

Figure 5.3: Minimal model with non-minimal branches.

(non-trivial) formulae that are true in the various time points, where a formula is not repeated if it was true earlier (so in the points labeled c and $\neg c$, also a and b are true). It can easily be checked that $\mathcal{M}$ is a model of T_Δ, and that it is minimal (if any true formulae are deleted anywhere, the result is not a model of T_Δ). However, the lower branch is not a minimal linear time model of T_Δ. This can be seen either by considering the smaller model where b is deleted from the second point onwards (which would still be a model of T_Δ), or by verifying that its limit model (in which a, b and $\neg c$ are known), does not correspond to an extension of Δ. The equivalence of these two methods follows from Theorem 5.15.

Given the set of linear time minimal models $\mathsf{MLT}(\Delta)$ of a temporal interpretation T_Δ of a default theory, these models can be jointly embedded in their coproduct $*\mathsf{MLT}(\Delta)$, which also is a minimal model of T_Δ. This provides one model to describe the complete semantics of the default theory. However, this model may contain a lot of redundant information: all branches at least have the same starting point, but in the coproduct a copy is included of this (actually identical information state) starting point for every branch. Moreover, branches can contain longer initial subsequences that are identical. In a coproduct these are not shared but present in a copy for each of the branches. A more compact form of a joint embedding of the minimal linear time models can be obtained by taking the closure of this coproduct, that is, by taking the joint closure of $\mathsf{MLT}(\Delta)$.

Definition 5.22 The joint closure $\mathrm{jcl}(\mathsf{MLT}(\Delta))$ of $\mathsf{MLT}(\Delta)$ is shortly denoted by LT^*_Δ.

Theorem 5.23

1. Let Δ be a default theory and $\mathfrak{S} \subseteq \mathsf{MLT}(\Delta)$ a set of minimal linear time models of T_Δ. Then the joint closure $\mathrm{jcl}(\mathfrak{S})$ of $\mathfrak{S}$ is a closed minimal temporal model of T_Δ. If $\mathfrak{S}$ is closed (in (LT, d)), then $\mathsf{Br}(\mathrm{jcl}(\mathfrak{S})) = \mathfrak{S}$.

2. The statement of item 1 holds in particular for the set $\mathsf{MLT}(\Delta)$ of all minimal linear time models of T_Δ: the model LT^*_Δ is a minimal model of T_Δ and if Δ is extension complete, then $\mathsf{Br}(\mathsf{LT}^*_\Delta) = \mathsf{MLT}(\Delta)$.

Proof: The joint closure $\mathrm{jcl}(\mathfrak{S})$ is closed by definition, and the (unique) homomorphism f mapping $*\mathfrak{S}$ into $\mathrm{jcl}(\mathfrak{S})$ is surjective, so $\mathrm{jcl}(\mathfrak{S})$ is a minimal model of T_Δ by Theorem 5.21, second part. Now suppose $\mathfrak{S}$ is closed. Using a similar argument as in the proof of Theorem 5.21, second part, one can show that every branch of $\mathrm{jcl}(\mathfrak{S})$ can be approximated by elements of $\mathfrak{S}$, which is closed and therefore contains such a branch. If Δ is extension complete, then by definition $\mathsf{MLT}(\Delta)$ is closed. □

The aim of this section was to find a branching time model containing just the Reiter extensions of Δ as limits of its branches. The following theorem shows that for an extension complete default theory Δ the model LT^*_Δ indeed fulfills this requirement.

Theorem 5.24 Let $\Delta = \langle D, W \rangle$ be an extension complete default theory with W consistent.

1. For every minimal linear time model of T_Δ there is a unique homomorphism into LT^*_Δ; this homomorphism is injective.

2. There is a bijection from the set $\mathsf{Ext}(\Delta)$ of all Reiter extensions of Δ onto the set $\mathsf{Br}(\mathsf{LT}^*_\Delta)$ of branches of LT^*_Δ. More precisely, the mapping $\Psi : \mathsf{Ext}(\Delta) \to \mathsf{Br}(\mathsf{LT}^*_\Delta)$ defined by $\Psi(E) = (Mod(E_i))_{i \in \mathbb{N}}$, has the inverse $\Phi : \mathsf{Br}(\mathsf{LT}^*_\Delta) \to \mathsf{Ext}(\Delta)$ defined by $\Phi(\mathcal{B}) = Th(\lim_{\mathcal{B}} \mathsf{LT}^*_\Delta)$. Furthermore, for every $i \in \mathbb{N}$ it holds $\Phi(\mathcal{B})_i = Th(\mathcal{B}_i)$.

Proof:

1. Every model $\mathcal{M}$ of $\mathsf{MLT}(\Delta)$ is mapped by inclusion (which is a homomorphism) into $*\mathsf{MLT}(\Delta)$ which is mapped into LT^*_Δ. The composition of these two homomorphisms is again a homomorphism. If there are two homomorphisms f, g mapping $\mathcal{M}$ into LT^*_Δ, then $f[\mathcal{M}]$ and $g[\mathcal{M}]$ are two isomorphic branches in a closed model, so these images must coincide as an easy consequence of Proposition 4.32. But this means that f and g are equal. It can easily be checked that a homomorphism from a linear model is always injective. Note that extension completeness is not needed here.

2. From Theorem 5.23 we have that $\mathsf{Br}(\mathsf{LT}^*_\Delta) = \mathsf{MLT}(\Delta)$, and Theorem 5.15 established a bijection between $\mathsf{MLT}(\Delta)$ and $\mathsf{Ext}(\Delta)$.

□

For the existence of a closed temporal model containing as branches just the minimal linear time models of a given default theory, the condition of extension completeness is not only sufficient, but also necessary, as is shown in the following proposition.

Proposition 5.25 For any default theory Δ the following are equivalent:

1. Δ is extension complete.

2. There exists a closed model $\mathcal{M}$ with $\mathsf{Br}(\mathcal{M}) = \mathsf{MLT}(\Delta)$.

Proof: From 1 to 2 is easy: the required model is LT^*_Δ. For the other direction, suppose we have a closed model $\mathcal{M}$ with $\mathsf{Br}(\mathcal{M}) = \mathsf{MLT}(\Delta)$. Take a converging sequence $\{\mathcal{B}_1, \mathcal{B}_2, \dots\}$ of models in $\mathsf{MLT}(\Delta)$, with limit $\mathcal{B}$. The models in the sequence are all present as branches in $\mathcal{M}$, and as $\mathcal{M}$ is closed, if two models in the sequence have an initial common subbranch, then these are mapped onto the same subbranch in $\mathcal{M}$. Take any initial subbranch of $\mathcal{B}$, then we can find a model $\mathcal{B}_i$ with the same initial subbranch, the image of which is in $\mathcal{M}$. If we extend this initial subbranch by one point, then we can again find a model $\mathcal{B}_j$ with this initial subbranch. Its image in $\mathcal{M}$ then extends the image of the subbranch of $\mathcal{B}_i$, as $\mathcal{M}$ is closed. In this fashion we find that $\mathcal{B}$ is a branch of $\mathcal{M}$, and therefore is in $\mathsf{MLT}(\Delta)$. $\square$

So if a default theory Δ is not extension complete, then LT^*_Δ contains a branch which is not a member of $\mathsf{MLT}(\Delta)$. Such a branch does not correspond to an extension. This means that the use of a temporal model construction as introduced here heavily depends on the topological properties of the given default theory: constructions fulfilling the requirements we imposed are not possible for non-extension complete default theories. However, recall Proposition 5.18, stating that this can only occur in the case of an infinite set of defaults. For almost all applications of default logic, the condition of extension completeness is fulfilled due to finiteness of the set of defaults.

5.2.4 Semantic entailment relations

Minimal semantic entailment relations based on the model LT^*_Δ and on the minimal linear models can be defined.

Definition 5.26 Let Δ be an extension complete default theory, and let φ be a formula. Define

$$\begin{array}{lcl} \Delta \mathrel{|\!\!\approx}_{\mathsf{LT}} \varphi & \Leftrightarrow & \forall \mathcal{M}[\mathcal{M} \text{ is a minimal linear time model of } T_\Delta \Rightarrow \mathcal{M} \models \varphi], \\ \Delta \mathrel{|\!\!\approx}_{\mathsf{LT}^*} \varphi & \Leftrightarrow & \mathsf{LT}^*_\Delta \models \varphi. \end{array}$$

For a certain class of formulae we can give logical relations between these entailment relations. Using Theorem 4.26, the following is easy to establish.

Proposition 5.27 For any default theory Δ, its temporal interpretation T_Δ is backward persistent under any homomorphism.

The following theorem gives more precise connections between the two semantic consequence relations.

Theorem 5.28 Let T_Δ be the temporal interpretation of an extension complete default theory Δ and let φ be a propositional formula.

1. If φ is backward persistent (under injections), then

$$\Delta \mathrel{|\!\!\approx}_{\mathsf{LT}^*} \varphi \Rightarrow \Delta \mathrel{|\!\!\approx}_{\mathsf{LT}} \varphi.$$

2. If φ is propositional, then

$$\Delta \mathrel{|\!\!\approx}_{\mathsf{LT}^*} \forall\mathrm{F}(K\varphi) \Leftrightarrow \Delta \mathrel{|\!\!\approx}_{\mathsf{LT}} \forall\mathrm{F}(K\varphi).$$

Proof:

1. Suppose $\Delta \approx_{\mathsf{LT}^*} \varphi$ and let $\mathcal{M}$ be a minimal linear time model of T_Δ. By Theorem 5.24, first part, there is an injective homomorphism f mapping $\mathcal{M}$ into LT^*_Δ. Take any time point s of $\mathcal{M}$, then since $\mathsf{LT}^*_\Delta \models \varphi$, in particular $(\mathsf{LT}^*_\Delta, f(s)) \models \varphi$. As φ is backward persistent under injections, we have $(\mathcal{M}, s) \models \varphi$. This proves that $\mathcal{M} \models \varphi$, and therefore $\Delta \approx_{\mathsf{LT}} \varphi$.

2. If φ is propositional, it is easy to see that $\forall \mathrm{F}(K\varphi)$ is backward persistent under any homomorphism, using Theorem 4.26. So the left to right direction follows by part 1. For the other direction, by Theorem 5.23 we have that $\mathsf{Br}(\mathsf{LT}^*_\Delta) = \mathsf{MLT}(\Delta)$. Take a point s in LT^*_Δ, and a branch $\mathcal{B}$ through s. Then $\mathcal{B} \in \mathsf{MLT}(\Delta)$, so $(\mathcal{B}, s) \models \forall \mathrm{F}(K\varphi)$, which means there must be a point $s \ll t$ with $\mathcal{B}(t) \models K\varphi$. But then also $(\mathsf{LT}^*_\Delta)_t \models K\varphi$. As the branch was arbitrary, we have $(\mathsf{LT}^*_\Delta, s) \models \forall \mathrm{F}(K\varphi)$. This proves that $\mathsf{LT}^*_\Delta \models \forall \mathrm{F}(K\varphi)$, so $\Delta \approx_{\mathsf{LT}^*} \forall \mathrm{F}(K\varphi)$.

□

We will show in Theorem 5.29 how these formulae $\forall \mathrm{F}(K\varphi)$ are related to sceptical entailment.

The model LT^*_Δ of an extension complete default theory Δ gives an overview of both all possible reasoning paths from a default theory (the branches) and the resulting conclusion sets (the limit models). Therefore in principle it contains all information that is relevant for an intended semantics. As a special case, the sceptical and credulous entailment relations (see Definition 3.1) can be based on this model. We define $(\mathsf{LT}^*_\Delta)^\omega$ as the set of the limit models of all branches of LT^*_Δ, i.e., $(\mathsf{LT}^*_\Delta)^\omega = \{\lim_{\mathcal{B}} \mathsf{LT}^*_\Delta \mid \mathcal{B} \text{ is a branch of } \mathsf{LT}^*_\Delta\}$.

Theorem 5.29 Let Δ be an extension complete default theory, r the root of LT^*_Δ and let φ be a propositional formula.

1. The following are equivalent:
   (a) φ is a sceptical consequence of Δ.
   (b) $(\mathsf{LT}^*_\Delta)^\omega \models \varphi$.
   (c) $(\mathsf{LT}^*_\Delta, r) \models \forall \mathrm{F}(K\varphi)$.
   (d) $(\mathcal{L}, s) \models \forall \mathrm{F}(K\varphi)$ for every minimal linear time model $\mathcal{L}$ of Δ with root s.

2. The following are equivalent:
   (a) φ is a credulous consequence of Δ.
   (b) $\lim_{\mathcal{B}} \mathsf{LT}^*_\Delta \models \varphi$ for some branch $\mathcal{B}$.
   (c) $(\mathsf{LT}^*_\Delta, r) \models \exists \mathrm{F}(K\varphi)$.

(d) $(\mathcal{L}, s) \models \exists \mathrm{F}(K\varphi)$ for some minimal linear time model $\mathcal{L}$ of Δ with root s.

(e) $(\mathcal{L}, s) \models \forall \mathrm{F}(K\varphi)$ for some minimal linear time model $\mathcal{L}$ of Δ with root s.

Proof:

1. From Theorem 5.24, we know that the function $\Psi : \mathsf{Ext}(\Delta) \rightarrow \mathsf{Br}(\mathsf{LT}^*_\Delta)$ defined by $\Psi(E) = (Mod(E_i))_{i \in \mathbb{N}}$, is a bijection. Now for any propositional φ, we have that $\varphi \in E \Leftrightarrow \varphi \in \bigcup E_i \Leftrightarrow Mod(E_i) \models \varphi$ for some $i \Leftrightarrow \lim\ ((Mod(E_i))_{i \in \mathbb{N}}) \models \varphi$ (this uses the fact that the sets $Mod(E_i)$ are closed S5-models). From these facts, it is easy to see that 1a and 1b are equivalent. The equivalence of 1b and 1c is immediate. From Theorem 5.23 we know that $\mathsf{Br}(\mathsf{LT}^*_\Delta) = \mathsf{MLT}(\Delta)$, from which we get the equivalence of 1c and 1d.

2. These equivalences can be proved analogously to those in part 1. The equivalence of 2d and 2e is an easy consequence of the semantic definitions of $\exists$F and $\forall$F.

□

Using the model LT^*_Δ we can define many more different consequence relations. Sceptical and credulous entailment use the formulae $\forall \mathrm{F}(K\varphi)$ and $\exists \mathrm{F}(K\varphi)$, but our temporal language is much more expressive. We can check for instance whether a certain propositional formula is true in every branch at a point with depth less than 5 (with the formula $\forall \mathrm{G} \forall \mathrm{G} \forall \mathrm{G} \forall \mathrm{G}(K\varphi)$).

In the case of normal default theories, there are even stronger connections between linear minimal models, branching time minimal models and the joint closures of these classes. We will treat them in the next subsection.

5.2.5 The case of normal default theories

A *normal* default rule is a default rule of the form $(\alpha : \beta)/\beta$, and a default theory consisting of solely normal default rules is called a *normal* default theory. In [ET94b] we pointed out a branching time temporal semantics for the normal case only. Most of the results there follow as a special case of the general case in this subsection.

If Δ is normal, the minimal temporal models of T_Δ can be characterized completely by their branches.

Theorem 5.30 Let Δ be a normal default theory. Then $\mathcal{M}$ is a minimal temporal model of T_Δ if and only if $\mathsf{Br}(\mathcal{M}) \subseteq \mathsf{MLT}(\Delta)$.

Proof: The right to left direction is Theorem 5.21. Note that the counterexample for the other direction following this theorem (see Figure 5.3) is based on a default theory with a non-normal default. So let us prove the other direction. Suppose $\mathcal{M}$ is a minimal model of T_Δ but has a branch $\mathcal{B}$ which is not a minimal model of T_Δ.

Consider the homomorphism f mapping $\mathcal{B}$, seen as a linear time model, into $\mathcal{M}$. As $\mathcal{M} \models T_\Delta$ and T_Δ is backward persistent under homomorphisms (Proposition 5.27), we have that $\mathcal{B} \models T_\Delta$. As it is not a minimal model of T_Δ by assumption, there must exist a linear time model $\mathcal{N}$ of T_Δ such that $\mathcal{N} \prec^g_{br} \mathcal{B}$. Suppose $\mathcal{B}$ and $\mathcal{N}$ are based on the flow of time $s_0 < s_1 < s_2 < \ldots$. Let us consider the first point of time s_n (from the roots) at which $\mathcal{N}$ and $\mathcal{B}$ are different. If $\mathcal{N}(s_0) \prec \mathcal{B}(s_0)$, then define a new model $\mathcal{M}'$ based on the same flow of time as $\mathcal{M}$ but with $\mathcal{M}'(s_0) = \mathcal{N}(s_0)$ and $\mathcal{M}'(t) = \mathcal{M}(t)$ for all $t \neq s_0$. It can easily be checked that $\mathcal{M}'$ is a model of T_Δ and $\mathcal{M}' \prec^g_{br} \mathcal{M}$, which is impossible since $\mathcal{M}$ was minimal.

Now suppose $n > 0$ so $\mathcal{N}(s_i) = \mathcal{B}(s_i)$ for $i < n$ and $\mathcal{N}(s_n) \prec \mathcal{B}(s_n)$. Construct a model $\mathcal{M}'$ based on the same flow of time as $\mathcal{M}$ but with $\mathcal{M}'(s_n) = \mathcal{N}(s_n)$ and $\mathcal{M}'(t) = \mathcal{M}(t)$ for $t \neq s_n$. We will show that $\mathcal{M}' \models T_\Delta$. It is clear that $\mathcal{M}' \models W'$, as this is evaluated per time point, and both $\mathcal{M} \models W'$ and $\mathcal{N} \models W'$. To show that $\mathcal{M}' \models \mathit{Cons}$, it is sufficient to show that $\mathcal{M}'$ is conservative. The only interesting case is for a point $t \ll s_n$. But the path from s_n to the root of its component is unique (Observation 4.18), and as s_n lies on $\mathcal{B}$, it must be the case that $t = s_i$ for some $i < n$. Then we have $\mathcal{M}'(s_i) = \mathcal{N}(s_i) \preceq \mathcal{N}(s_n) = \mathcal{M}'(s_n)$, as $\mathcal{N}$ is conservative ($\mathcal{N} \models \mathit{Cons}$).

Now take a rule $P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta) \rightarrow K\beta$ (remember that Δ is normal). It is easy to see that if at a point in $\mathcal{M}$ the left hand side is false, it will also be false in the corresponding point of $\mathcal{M}'$ (this uses conservativity of $\mathcal{M}'$). So the only possibility of this rule to be false in $\mathcal{M}'$, is at time point s_n. We will show that this cannot occur. So suppose we have $(\mathcal{M}', s_n) \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$. As $\mathcal{M}'(t) = \mathcal{M}(t)$ for $t \neq s_n$ and the truth of the formula $P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$ does not depend on the information state at the current point in time, it easily follows that $(\mathcal{M}, s_n) \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$. This implies $(\mathcal{M}, s_n) \models K\beta$, so $(\mathcal{B}, s_n) \models K\beta$. This means (by conservativity of $\mathcal{B}$) that $(\mathcal{B}, s_i) \not\models K\neg\beta$ for all $i \in \mathbb{N}$. As $\mathcal{N} \prec^g_{br} \mathcal{B}$, we also have that $(\mathcal{N}, s_i) \not\models K\neg\beta$, so $(\mathcal{N}, s_n) \models \neg\forall\mathrm{F}(K\neg\beta)$. Since $(\mathcal{M}, s_n) \models P(K\alpha)$, we have $(\mathcal{B}, s_n) \models P(K\alpha)$, from which it follows that $(\mathcal{N}, s_n) \models P(K\alpha)$ ($\mathcal{N}(s_i) = \mathcal{B}(s_i)$ for $i < n$). So $(\mathcal{N}, s_n) \models P(K\alpha) \wedge \neg\forall\mathrm{F}(K\neg\beta)$. As $\mathcal{N} \models D'$, we get $(\mathcal{N}, s_n) \models K\beta$, and from $\mathcal{M}'(s_n) = \mathcal{N}(s_n)$ we conclude that $(\mathcal{M}', s_n) \models K\beta$. We have shown that $\mathcal{M}' \prec^g_{br} \mathcal{M}$ and $\mathcal{M}' \models T_\Delta$, which contradicts the assumption that $\mathcal{M}$ is a minimal model of T_Δ. This means that $\mathcal{B}$ must be a minimal model of T_Δ, so $\mathsf{Br}(\mathcal{M}) \subseteq \mathsf{MLT}(\Delta)$, which concludes the proof. □

For the case of closed models this implies the following characterization result of closed minimal temporal models.

Proposition 5.31 Suppose Δ is an extension complete normal default theory and $\mathcal{M}$ a temporal model. Then $\mathcal{M}$ is a closed minimal temporal model of T_Δ if and only if $\mathcal{M}$ is the joint closure of a set $\mathfrak{B}$ of minimal linear time models of T_Δ.

Proof: The joint closure of a set $\mathfrak{B}$ of minimal linear time models of T_Δ is a closed minimal temporal model of T_Δ by Theorem 5.23 (extension completeness and normality of Δ is not used). For the other direction, by Theorem 4.40, $\mathcal{M}$ is the joint closure of its branches. These branches are minimal linear time models of T_Δ by Theorem 5.30. $\square$

For extension complete normal default theories, the model LT^*_Δ also has stronger properties.

Definition 5.32 (Final minimal model) The model $\mathcal{F}$ is called a *final minimal temporal model* of T_Δ if it is a final model in the class of minimal models of T_Δ (see Subsection 4.3.4, below Theorem 4.40), that is, if it is a minimal temporal model of T_Δ and for each minimal temporal model $\mathcal{M}$ of T_Δ there is a unique homomorphism $f : \mathcal{M} \to \mathcal{F}$.

We have the following result showing that in the normal case, we have a final model semantics for default logic:

Theorem 5.33 Let Δ be a normal extension complete default theory. Then LT^*_Δ is a (unique) final minimal temporal model of T_Δ, so for every minimal temporal model of T_Δ there is a unique homomorphism into LT^*_Δ; for closed minimal temporal models of T_Δ this homomorphism is injective. The model LT^*_Δ is also the joint closure of all minimal temporal models of T_Δ.

Proof: By definition, LT^*_Δ is the joint closure of $\mathsf{MLT}(\Delta)$, so using Proposition 5.31 we have that it is a minimal temporal model of T_Δ. Now consider any minimal temporal model $\mathcal{M}$ of T_Δ. By Theorem 5.30 it follows that $\mathsf{Br}(\mathcal{M}) \subseteq \mathsf{MLT}(\Delta)$, and from Theorem 5.23 it follows that $\mathsf{Br}(\mathsf{LT}^*_\Delta) = \mathsf{MLT}(\Delta)$. The required unique homomorphism maps every branch of $\mathcal{M}$ into its (unique) place in LT^*_Δ. This uniqueness follows from the closedness of LT^*_Δ. By Definition 4.31, any homomorphism from a closed model is injective. $\square$

When a default theory is normal and extension complete, semantics can be defined by taking all linear minimal models of T_Δ, by taking all branching time minimal models of T_Δ, or by taking the unique final minimal temporal model, LT^*_Δ, that incorporates all these possibilities in the most efficient manner.

A theory T_Δ specifies default reasoning using the default theory Δ. It gives rise to a reasoning frame operator, but we can not directly use Definition 4.63, since it was geared towards linear time temporal logic. But as a branching time model can be seen as a (compact) representation of traces (this role is played by the branches), such an operator is easy to define:

$$\mathcal{T}_{T_\Delta}(X) = \{\mathcal{B} \mid \mathcal{B} \in \mathsf{Br}(\mathsf{LT}^*_\Delta) \text{ and } Th(\mathcal{B}_0) = Cn(X)\}.$$

In the case of normal default theories, the definition is more like Definition 4.63, as we then have

$$\mathcal{T}_{T_\Delta}(X) = \{\mathcal{B} \mid \mathcal{B} \in \mathsf{Br}(\mathcal{M}), \mathcal{M} \models_{\preceq^g_{br}} T_\Delta \text{ and } Th(\mathcal{B}_0) = Cn(X)\}.$$

Let us briefly review what we have done in this section. We described the construction of a branching time temporal model in which all minimal linear time models are incorporated. Under a topological condition (extension completeness), which is always satisfied for finite default theories, this model contains only branches which are minimal linear time models. For any normal default theory satisfying the same condition, this model contains not only all minimal linear models of the temporal interpretation, but also all minimal branching time models. In this case we have a linear, branching time and final model semantics for default logic.

5.3 Logic programming

In Section 3.2, we introduced positive logic programs and their semantics and associated reasoning frame operator. In this section, we will show that logic programming can be specified in temporal logic. Many extensions of the basic paradigm of minimal models of positive programs exist; we will consider the semantics of *stable generated models* of extended generalized logic programs, introduced in [HW97b].

A logic program consists of *facts* and *deduction rules*. Facts correspond to sentences of a suitably restricted language, and deduction rules correspond to non-schematic (Gentzen) sequents. While facts express extensional knowledge, rules express intensional knowledge. A set of facts can be viewed as a database whose semantics is determined by its minimal models: a fact is true if it is a member of this set, and false otherwise. In the case of logic programs, minimal models are not adequate because they are not able to capture 'groundedness', i.e. the directedness of rules. Therefore, *stable* models in the form of certain fixpoints have been proposed by Gelfond and Lifschitz ([GL88]) as the intended models of normal logic programs. In [HJW97], this notion was generalized by presenting a definition which is neither fixpoint-based nor dependent on any specific rule syntax: *stable generated models*.

We will start by introducing stable generated models, after which we will study the relation with our temporal semantics.

5.3.1 Preliminaries

Our presentation of stable generated models will be (slightly) different from the presentation in [HW97b]. In logic programming, rules are usually allowed to contain (free) variables. In most treatments however, including the one based on stable generated models, semantically a rule with free variables is equivalent to the set of ground instantiations of the rule. A ground instantiation of a rule is the result of substituting ground terms from the Herbrand base for all of its free variables; the Herbrand base of a program is the smallest set containing all constants occuring in the program and closed under formation of terms using function symbols occuring in the program. In view of this equivalence we will give a purely propositional presentation. The models used in [HW97b], are Herbrand interpretations, which are essentially just sets of literals. The definition of stable generated models uses only *coherent* Herbrand interpretations, in which an atom and its negation cannot both occur. As such, a coherent Herbrand interpretation I is isomorphic to a partial model m via the correspondence

$$\forall l \in Lit(P) : (l \in I \Leftrightarrow m(l) = 1)$$

where $Lit(P) = P \cup \{\neg a \mid a \in P\}$. We will use partial models in our presentation.

To the propositional language we add one more connective, namely *weak negation*, which will be denoted as $-$. The interpretation of weak negation in a partial model is defined by the following table:

$-$	
0	1
1	0
u	1

Program rules are presented as sequents.

Definition 5.34 (Sequent) A *sequent* s is an expression of the form

$$F_1, \ldots, F_m \;\Rightarrow\; G_1, \ldots, G_n$$

where F_i, G_j are formulae in the propositional language extended with weak negation, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. The *body* of s, denoted by Bs, is given by $\{F_1, \ldots, F_m\}$, and the *head* of s, denoted by Hs, is given by $\{G_1, \ldots, G_n\}$. *Seq* denotes the set of all sequents.

For a sequent $\Rightarrow F$ with empty body we also write more simply F.

Define $XLit(P) = Lit(P) \cup \{-l \mid l \in Lit(P)\}$. We will often drop the argument P. A number of classes of sequents will be defined. The most general class for which stable generated models are defined is EGLP (*extended general logic programs*). The class ELP is the class of rules in extended logic programs (the word *extended* means that one can use, besides negation by failure, also classically negated, or strongly negated, atoms). The class EDLP allows disjunctive conclusions. More formally, define:

1. EGLP $= Seq$.
2. EDLP $= \{s \in Seq \mid Bs \subseteq XLit, Hs \subseteq Lit\}$.
3. ELP $= \{s \in Seq \mid Bs \subseteq XLit, Hs \subseteq Lit, \#Hs = 1\}$.

A sequent s is true in a partial model m, denoted $m \models s$, if $m(\bigvee Hs) = 1$ whenever $m(\bigwedge Bs) = 1$. The set of partial models of a set of sequents S, denoted

$Mod(S)$, is defined by $Mod(S) = \{m \in \mathsf{IS}^{3val} \mid m \models s \text{ for all } s \in S\}$. With respect to a *set* of partial models M, we write $M \models F$ iff $m(F) = 1$ for all $m \in M$. We define the set S_M of all sequents from a sequent set S which are applicable in M by

$$S_M = \{s \in S \mid M \models \bigwedge Bs\}$$

We will now define a semantics for logic programs based on [HW97b].

5.3.2 Stable generated models

A sequent s may have several meanings. The traditional meaning of a sequent, based on classical model theory, is given by the formula $\bigwedge Bs \to \bigvee Hs$ and the usual model relation. Our intuitive understanding of rules suggests another meaning which interprets a sequent as a rule for generating factual (extensional) knowledge. We consider a model of a set S of sequents as intended if it can be generated bottom-up starting from zero information by an iterated application of the sequents $s \in S$. This intention is captured by notion of a stable generated model. The subsequent definition is a generalization of the notion discussed in [HW97b]. This notion generalizes the answer set semantics of [GL90]. In the sequel, we need the notion of an 'interval' of partial models. For two partial models m_1 and m_2, we define $[m_1, m_2] = \{n \in \mathsf{IS}^{3val} \mid m_1 \preceq n \preceq m_2\}$, where $\preceq$ is the information ordering on IS^{3val}. Furthermore, if $\langle Y, < \rangle$ is a partial order, then $Min(Y, <)$ denotes the set of all minimal elements of Y, i.e. $Min(Y, <) = \{x \in Y \mid \text{ there is no } x' \in Y \text{ such that } x' < x\}$.

Definition 5.35 (Stable generated model) Let $S \subseteq EGLP$. A partial model m is called a *stable generated model* of S if there is an ordinal κ and a chain of partial models $n_0 \preceq n_1 \preceq \ldots \preceq n_\kappa$ such that $m = n_\kappa$, and

1. $n_0(p) = u$ for all $p \in P$.
2. For successor ordinals α with $0 < \alpha \leq \kappa$, n_α is a $\preceq$-minimal $\preceq$-extension of $n_{\alpha-1}$ satisfying the heads of all sequents whose bodies hold in $[n_{\alpha-1}, m]$, i.e.,

   $$n_\alpha \in Min(\{n \in \mathsf{IS}^{3val} \mid n_{\alpha-1} \preceq n, \text{ and } n(\bigvee Hs) = 1, \text{ for all } s \in S_{[n_{\alpha-1},m]}\}, \preceq).$$

3. For limit ordinals $\lambda \leq \kappa$, $n_\lambda = \sup_{\preceq}\{n_\alpha \mid \alpha < \lambda\}$.

We say that n is *generated* by the *S-stable chain* $n_0 \preceq \ldots \preceq n_\kappa$.

The stable entailment relation is defined as follows:

$$S \models^s_c F \Leftrightarrow m(F) = 1 \text{ for all stable generated models } m \text{ of } S.$$

The definition of a stable chain allows chains of any length. However, this is not necessary.

Proposition 5.36 ([HW97b]) Every stable chain can be shortened or extended to a chain isomorphic to the natural numbers, generating the same stable model.

For purposes of comparison, we also give the definition of the answer set semantics of [GL91], generalizing the stable model semantics of [GL88]. We use the presentation with sets of literals here.

For $B \subseteq XLit$, let B^- denote the set of literals which occur weakly negated in B, i.e., $B^- = \{k \in Lit \mid -k \in B\}$, and let $B^+ = \{k \in Lit \mid k \in B\}$.

Definition 5.37 Let $I \subseteq Lit$, and $S \subseteq EDLP$. Then the *Gelfond-Lifschitz transformation* of S with respect to I is defined as

$$S^I = \{B^+ \Rightarrow H \mid (B \Rightarrow H) \in S, \text{ and } B^- \cap I = \emptyset\}$$

and the *generalized Gelfond-Lifschitz operator* Γ_S is defined as follows: $\Gamma_S(I)$ collects all minimal models of S^I, i.e. $\Gamma_S(I) = Min(Mod(S^I), \subseteq)$. A set $I \subseteq Lit$ is called an *answer set* of S, if $I \in \Gamma_S(I)$.

On (non-disjunctive) extended logic programs, answer sets correspond to stable generated models, with the exception of the inconsistent answer sets.

Proposition 5.38 Let $S \subseteq ELP$. If I is a coherent answer set, then the partial model m defined by $m(l) = 1 \Leftrightarrow l \in I$ is a stable generated model of S. If m is a stable generated model of S, then the set $I = \{l \in Lit \mid m(l) = 1\}$ is an answer set of S.

Proof: The statement is proved in [HW97b] for normal logic programs (in which no classical negation is used). It is straightforward to generalize this to extended programs. □

For disjunctive programs, with or without classical negation, stable generated models do *not* generally coincide with answer sets, as the following example shows.

Example 5.39 Let $S = \{\Rightarrow a, b;\ a \Rightarrow b;\ -a \Rightarrow a\}$. Let m_0 be the model in which $m_0(a) = m_0(b) = u$, m_a the model in which $m(a) = 1$ and $m(b) = u$, and m_{ab} the one in which $m(a) = m(b) = 1$. We show that m_{ab} is a stable generated model of S by giving a stable chain for it. The chain must start with m_0. Then $S_{[m_0, m_{ab}]} = \{\Rightarrow a, b\}$. One of the two minimal extensions of m_0 is m_a which we take as second model in the chain. Then $S_{[m_a, m_{ab}]} = \{a \Rightarrow b; \Rightarrow a, b\}$, and a minimal extension of m_a satisfying b gives the final model m_{ab}. On the other hand, $\{a, b\}$ is not a minimal model of the Gelfond-Lifschitz reduct of S with respect to $\{a, b\}$, which is $\{\Rightarrow a, b; a \Rightarrow b\}$, since $\{b\}$ is a model of it. In fact, S has no answer sets.

We recall some notions and results from [HW97a] that will be used in the rest of this section. Two programs P and Q are said to be *stable equivalent*, denoted as $P \equiv_{st} Q$, if P and Q have the same stable generated models. A rule $s \in$ EGLP has *normal form* if $s = F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$, where every F_i is a disjunction of literals from $XLit$, and every G_j is a literal from $XLit$. The sequent s has *special normal* form if it has normal form and every F_i is a literal. In [HW97a] it is shown that every program can be transformed into a stable equivalent program whose rules have normal form. Let $Fin(Seq) = \{X : X \subseteq Seq$ and X is finite $\}$.

Definition 5.40 (Transformation rule, [HW97a]) A relation $r \subseteq Fin(Seq) \times Fin(Seq)$ is said to be an admissible transformation rule if r is decidable and for every program $P \subseteq EGLP$, and $s \in P$, $\{X, Y\} \subseteq Fin(Seq)$ such that $r(Y, X)$ the condition $P \equiv_{st} (P - Y) \cup X$ is satisfied. Let $R = \{r_1, ..., r_m\}$ be a finite set of admissible transformation rules, and P, Q programs. The relation $\longrightarrow_R$ between logic programs is defined as follows: $P \longrightarrow_R Q$ iff there is a rule $r \in R$, $Y \subseteq P$, $X, Y \in Fin(Seq)$ such that $r(Y, X)$ and $Q = (P - Y) \cup X$. Let $\longrightarrow_R^\star$ be the transitive closure of $\longrightarrow_R$. P can be transformed into Q by the rule system R if $P \longrightarrow_R^\star Q$.

Lemma 5.41 ([HW97a]) Let $P \subseteq EGLP$, and $s \in P$, $F \equiv \bigwedge Bs$, and $G \equiv \bigvee Hs$. Then $P \equiv_{st} (P - \{s\}) \cup \{Bs \Rightarrow G\} \equiv_{st} (P - \{s\}) \cup \{F \Rightarrow Hs\}$.

Lemma 5.42 ([HW97a]) Let $P \subseteq GLP$, and $s \in P$, $s = Bs \Rightarrow F_1 \wedge F_2$. Then $P \equiv_{st} (P - \{s\}) \cup \{Bs \Rightarrow F_1, Bs \Rightarrow F_2\}$.

Corollary 5.43 ([HW97a]) There is a finite set R of admissible transformation rules such that for every finite extended general logic program P there exists a program Q in normal form such that $P \longrightarrow_R^\star Q$.

Corollary 5.44 ([HW97a]) For every finite extended general logic program P there exists a program Q in normal form such that $P \equiv_{st} Q$.

From here on, we will also assume that for a rule $F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$, no F_i contains a pair $l, -l$. Such an F_i is a tautology, and can be eliminated from the body altogether (without changing the semantics).

5.3.3 A temporal interpretation of logic programming

We would like to specify the behavior of an agent reasoning with a logic program. That is, as was the case for default logic, we are looking for a translation of logic programs into a theory of temporal logic. Since a logic program only allows us to derive literals, we will use partial logic. We first prove a lemma.

Lemma 5.45 For a rule $F_1, \ldots, F_p \Rightarrow G_1, \ldots G_q$ in normal form, and partial models n and m, we have:
$[n,m] \models F_1, \ldots, F_p \Leftrightarrow$ for all $1 \le i \le p$ there is $l_i \in F_i : l_i \in Lit$ and $n(l_i) = 1$, or $l_i = -k$ for some $k \in Lit$ and $m(k) \neq 1$.

Proof: "$\Leftarrow$": Let j be a partial model such that $n \preceq j \preceq m$, and let $1 \le i \le p$. Suppose there is an l_i in F_i with $l_i \in Lit$ and $n(l_i) = 1$, then by persistence (see Proposition 2.4) $j(l_i) = 1$. Otherwise there is an l_i in F_i with $l_i = -b$ (and $b \in Lit$) and $m(b) \neq 1$. But then $j(b) \neq 1$ so $j(l_i) = 1$. This means that $j(F_i) = 1$ so $j(F_1 \wedge \cdots \wedge F_p) = 1$. We conclude that $[n,m] \models F_1 \wedge \cdots \wedge F_p$.

"$\Rightarrow$": Suppose there is an F_i with $\forall l \in F_i$: if $l \in Lit$ then $n(l) \neq 1$ and if $l = -b$ with $b \in Lit$ then $m(b) = 1$. Define the partial model j by setting, for $l \in Lit$:

$$j(l) = \begin{cases} n(l) & \text{if } n(l) \neq u \\ 1 & \text{if } -l \in F_i \\ u & \text{otherwise.} \end{cases}$$

One can easily check that $n \preceq j \preceq m$. Now take $l \in F_i$. If $l \in Lit$ then $-l \notin F_i$, so $j(l) = n(l) \neq 1$. Otherwise, $l = -b$ for $b \in Lit$, so $j(b) = 1$, and $j(l) \neq 1$. Thus, $j(F_1 \wedge \cdots \wedge F_p) \neq 1$, whence $[n,m] \not\models F_1 \wedge \cdots \wedge F_p$. $\square$

This lemma suggests a translation for a sequent in normal form. A disjunction F_i should be translated as a disjunction. The disjunction is true if there is a literal true in the current point, or if there is a weakly negated literal which is not true at the limit state. The formal definition follows below.

Definition 5.46 Let $s \in EGLP$ be in normal form, say $s = F_1, \ldots, F_n \Rightarrow G_1, \ldots, G_m$, with $F_i = b_1^i \vee \cdots \vee b_{l(i)}^i \vee -c_1^i \vee \cdots \vee -c_{m(i)}^i$ and $G_j = d_j$ for $1 \le j \le j_0$ for a j_0 with $0 \le j_0 \le m$ and $G_j = -e_j$ for $j_0 < j \le m$, where each b_i, c_i, d_i, and e_i is a member of Lit and $l(i), m(i), k, j \ge 0$. Define the function ρ as follows

$$\rho(F_i) = Cb_1^i \vee \cdots \vee Cb_{l(i)}^i \vee \neg FCc_1^i \vee \cdots \vee \neg FCc_{m(i)}^i,$$

$$\rho(d_i) = XCd_i,$$

$$\rho(-e_i) = \neg XCe_i, \text{ and}$$

$$\rho(s) = \rho(F_1) \wedge \cdots \wedge \rho(F_n) \rightarrow \rho(G_1) \vee \cdots \vee \rho(G_m).$$

For $P \subseteq EGLP$ in normal form, $\rho(P) = \{\rho(s) \mid s \in P\}$.

The following proposition shows that the translation is faithful (e.g., it preserves semantics).

Proposition 5.47 Let $P \subseteq EGLP$ be in normal form, then

1. For every stable generated model n of P, generated by the chain $n_0 \preceq n_1 \preceq \dots$, the temporal partial model $(M_t)_{t\in\mathbb{N}}$, defined by $M_t = n_t$ for all $t \in \mathbb{N}$, is a $\preceq^g$-minimal model of $\rho(P)$, and $\lim M = n$.
2. For every minimal temporal partial model $(M_t)_{t\in\mathbb{N}}$ of $\rho(P)$, the model $\lim M$ is a stable generated model of P, generated by the stable chain $M_0, M_1, \dots$.

Proof: Suppose we have a sequence $(M_t)_{t\in\mathbb{N}}$ of partial models. Then obviously, this can be seen either as a chain of models, or as a temporal partial model. If we can show that the requirements for being a P-stable chain are equivalent to the requirements for being a minimal temporal partial model of $\rho(P)$, then we are done. In order to show this, we need the following representation result for MTPL, analogous to Proposition 4.54. The notions of input and reasoning formulae for MTPL are the same as those for MTEL, when the K-operator is replaced by a C-operator. Now let T be a theory consisting of input formulae and reasoning formulae of MTPL. Then for any conservative TPL-model $\mathcal{M}$, the following holds:

$$\mathcal{M} \models_{\preceq^g} T$$

$$\Leftrightarrow$$

1. $\mathcal{M}_0$ is a $\preceq$-minimal model of the input formulae in T.
2. For each $i \in \mathbb{N}$, $\mathcal{M}_{i+1}$ is a $\preceq$-minimal element of the set of partial models which are $\preceq$-extensions of $\mathcal{M}_i$ satisfying the conclusions of the reasoning formulae of T applicable in $\mathcal{M}$ at time point i.

We will not give a proof here, since it is analogous to the proof of Proposition 4.54.

It is easy to see that $\rho(P)$ does not contain input formulae, so that $\mathcal{M}_0$ must be a $\preceq$-minimal model of the empty set, which means it must assign u to all atoms, which is the first requirement for a stable chain.

For a rule s, it is also straightforward to check that a partial model satisfies $\bigvee Hs$ if and only if it satisfies the conclusion of $\rho(s)$ (in the sense corresponding to Definition 4.53). Now suppose $s = F_1, \dots, F_n \Rightarrow G_1, \dots, G_m$, and for some $j_0 \in \mathbb{N}$, $\rho(s)$ is applicable in $(M_t)_{t\in\mathbb{N}}$ at j_0. This means that $(M, j_0) \models \rho(F_1) \wedge \dots \wedge \rho(F_n)$. Let $1 \leq i \leq n$ be arbitrary. Then $(M, j_0) \models \rho(F_i)$. If $F_i = b_1^i \vee \dots \vee b_{l(i)}^i \vee -c_1^i \vee \dots \vee -c_{m(i)}^i$, then $(M, j_0) \models Cb_1^i \vee \dots \vee Cb_{l(i)}^i \vee \neg FCc_1^i \vee \dots \vee \neg FCc_{m(i)}^i$. Now if $(M, j_0) \models Cb_j^i$ for some j, then $b_j^i \in F_i$ and $b_j^i \in Lit$ and $M_{j_0}(b_j^i) = 1$. If not, then $(M, j_0) \models \neg FCc_j^i$ for some j. But this means that $\lim M(c_j^i) \neq 1$, whereas $-c_j^i \in F_i$ and $c_j^i \in Lit$. Now Lemma 5.45 gives us that $[M_{j_0}, \lim M] \models F_1, \dots, F_n$. Likewise, we can prove that if $[M_{j_0}, \lim M] \models F_1, \dots, F_n$, then $(M, j_0) \models \rho(F_1) \wedge \dots \wedge \rho(F_n)$. But this means that the second requirement above is equivalent to the second requirement for being a stable chain. $\square$

We will give an example.

Example 5.48 (Continued example) The temporal translation of Example 5.39 is:

$$\{XCa \vee XCb, Ca \rightarrow XCb, \neg FCa \rightarrow XCa\}.$$

This theory has only one minimal model:

		time				
		0	1	2	3	...
atoms	a	u	1	1	1	...
	b	u	u	1	1	...

It is easy to see that this model corresponds to the stable generated model of the original program and to the stable chain generating it.

So logic programming (with stable generated semantics) can be specified in temporal logic. For a logic program P in normal form, the temporal theory $\rho(P)$ induces a reasoning frame operator according to Definition 4.63:

$$\mathcal{T}_{\rho(P)}(X) = \{M \mid \quad M \text{ is a temporal partial model and } M \models_{\preceq^g} \rho(P) \text{ and } \mathrm{Th}(M_0) = X\}.$$

In fact, $\mathcal{T}_{\rho(P)}(X)$ is only non-empty for $X = \emptyset$.

The translation into temporal logic simplifies the proof of the following proposition, which continues the discussion on normal forms at the end of Section 5.3.2.

Proposition 5.49 For every extended general logic program P, there exists a stable equivalent program Q which is in special normal form, i.e., all its rules are of the form $F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$ where the F_i and G_j are in $XLit$.

Proof: Let $P \subseteq EGLP$. By Corollary 5.44, P can be transformed into a stable equivalent program P' in normal form. Then we can delete, for every rule, any F_i in the body which contains a complementary literal pair $\{l, -l\}$ with $l \in Lit$. Let the result be P''. Now consider any rule $s \in P''$, say $s = F_1, \ldots, F_m \Rightarrow G_1, \ldots, G_n$, and suppose that $F_k = b_1 \vee \ldots \vee b_{l(k)} \vee -c_1 \vee \ldots \vee -c_{m(k)}$ for some $1 \leq k \leq m$. Then $\rho(s) = \rho(F_1) \wedge \ldots \wedge \rho(F_m) \rightarrow \rho(G_1) \vee \ldots \vee \rho(G_n)$ and $\rho(F_k) = Cb_1 \vee \ldots \vee Cb_{l(k)} \vee \neg FCc_1 \vee \ldots \vee \neg FCc_{m(k)}$. But in temporal partial logic, the connectives behave classically: for any temporal model $\mathcal{M}$ and $t \in \mathbb{N}$ it holds that $(\mathcal{M}, t) \models \alpha \wedge (\beta \vee \gamma) \rightarrow \delta$ if and only if $(\mathcal{M}, t) \models \alpha \wedge \beta \rightarrow \delta$ and $(\mathcal{M}, t) \models \alpha \wedge \gamma \rightarrow \delta$.

This means that we may replace $\rho(s)$ by the rules

$$\begin{array}{lcl}
\rho(F_1) \wedge \ldots \wedge Cb_1 \wedge \ldots \wedge \rho(F_m) & \rightarrow & \rho(G_1) \vee \ldots \vee \rho(G_n), \\
 & \vdots & \\
\rho(F_1) \wedge \ldots \wedge Cb_{l(k)} \wedge \ldots \wedge \rho(F_m) & \rightarrow & \rho(G_1) \vee \ldots \vee \rho(G_n), \\
\rho(F_1) \wedge \ldots \wedge \neg FCc_1 \wedge \ldots \wedge \rho(F_m) & \rightarrow & \rho(G_1) \vee \ldots \vee \rho(G_n), \\
 & \vdots & \\
\rho(F_1) \wedge \ldots \wedge \neg FCc_{m(k)} \wedge \ldots \wedge \rho(F_m) & \rightarrow & \rho(G_1) \vee \ldots \vee \rho(G_n).
\end{array}$$

Let T be the temporal theory obtained from $\rho(P'')$ by performing the above operation until there are no more disjunctions on the left hand side of any implication. Then T contains only formulae of the form $Ca_1 \wedge \ldots \wedge Ca_n \wedge \neg FCb_1 \wedge \ldots \wedge \neg FCb_m \rightarrow XCd_1 \vee \ldots \vee XCd_k \vee \neg XCe_1 \vee \ldots \vee \neg XCe_l$. But such a formula is the translation of the rule $a_1, \ldots, a_n, -b_1, \ldots, -b_m \Rightarrow d_1, \ldots, d_k, -e_1, \ldots, -e_l$. This means we can find a program Q such that $\rho(Q)$ is exactly T. This Q has special normal form. Furthermore, P is stable equivalent to P', which in turn is equivalent to P''. Stable models of P'' correspond to minimal temporal models of $\rho(P'')$, which is equivalent to T (and therefore has the same minimal temporal models). But $T = \rho(Q)$, so the minimal temporal models of T correspond to stable generated models of Q. We conclude that Q is stable equivalent to P. □

We have shown that logic programs can be translated (via a normal form) to formulae of minimal temporal partial logic in a modular way (the translation of a program is the union of the translations of its sequents). The translation preserves the semantics (in the sense of Proposition 5.47). But how about the other direction, i.e., is there a modular translation of minimal temporal partial logic into logic programs (endowed with the stable generated semantics)? For this question to be answered positively, it is sufficient (and necessary) that temporal formulae in a normal form can be translated. First of all, formulae with nested temporal operators are in general not expressible in logic programming. The formula $XXXCa$ has one minimal temporal model, which assigns 1 to a from time point 3 onwards, and u to a before 3, and u to any other literal, at any point in time. It is not hard to see that this model can not correspond to a stable generated model of a program. Let us restrict the language by only considering formulae which are built up, using boolean connectives, from formulae of the form $XC\alpha, GC\alpha, FC\alpha, C\alpha$, where α is propositional. Essentially, the temporal language is then a propositional language using the above formulae as its propositional atoms. Therefore, any temporal formula is equivalent to a formula in conjunctive normal form, and we may focus on the disjuncts. As only conservative models are considered, it is the case that $GC\alpha$ is equivalent to $XC\alpha$. We may thus concentrate on disjunctions containing $C\alpha$, $\neg C\alpha$, $FC\alpha$, $\neg FC\alpha$, $XC\alpha$, and $\neg XC\alpha$. It can be checked that the C-operator distributes over conjunction and disjunction, e.g. $C(\alpha \wedge \beta)$ is equivalent to $C\alpha \wedge C\beta$, and $C(\alpha \vee \beta)$ is equivalent to $C\alpha \vee C\beta$. This means that an atom $C\alpha$ is equivalent to $C\alpha'$, where α' is a conjunctive normal form (or disjunctive normal form) of α. We may then distribute the C operator over the conjunctions and disjunctions. Furthermore, in conservative models the temporal operators distribute over conjunctions and disjunctions of $C\alpha$ formulae. So we can even restrict ourselves to disjunctions of formulae of the form Ck, $\neg Ck$, FCk, $\neg FCk$, XCk, and $\neg XCk$, where k is a literal. Any disjunction not containing atoms of the form Ck or $\neg FCk$, and not containing $\neg Ck$ and FCk for the same literal k, can be translated faithfully to a sequent. Consider a disjunction of the following form (for clarity, we include only one of each sort of temporal atom):

$$\neg Ca \vee FCb \vee XCc \vee \neg XCd$$

where $a, b, c, d \in Lit$. This is equivalent to the implication

$$Ca \wedge \neg FCb \longrightarrow XCc \vee \neg XCd,$$

which is just the translation of the sequent

$$a, -b \Rightarrow c, d.$$

So how about the other two sorts of atoms? An easy example of a formula that cannot be translated is Ca. Its only minimal temporal partial model assigns 1 to a and u to any other atoms, for every time point. However, any stable chain starts with the empty set. So, no logic program has a stable chain equivalent to this minimal temporal partial model. One might think that the only formulae that cannot be translated, are formulae that have a minimal temporal partial model $\mathcal{M}$ for which $\mathcal{M}_0$ does not correspond to the empty set. This is not true, however, as witnessed by the following set of formulae:

$$\begin{array}{ll}
(1) & XCa \\
(2) & Ca \rightarrow XCb \\
(3) & Ca \wedge Cb \rightarrow XCc \\
(4) & Ca \wedge \neg Cb \rightarrow XCd \\
(5) & Ca \wedge Cb \wedge Cc \rightarrow XCd.
\end{array}$$

The conjunction of these formulae has two minimal models:

$$\mathcal{M}: \begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & \ldots \\ \hline
a & u & 1 & 1 & 1 & \ldots \\
b & u & u & 1 & 1 & \ldots \\
c & u & u & u & 1 & \ldots \\
d & u & u & 1 & 1 & \ldots
\end{array}
\qquad
\mathcal{N}: \begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & \ldots \\ \hline
a & u & 1 & 1 & 1 & \ldots \\
b & u & 1 & 1 & 1 & \ldots \\
c & u & u & 1 & 1 & \ldots \\
d & u & u & u & 1 & \ldots
\end{array}$$

These models cannot both correspond to stable chains of the same program. Since they have the same limit model, the same sequents are applicable with respect to $[\emptyset, Lit(\lim \mathcal{M})]$ as with respect to $[\emptyset, Lit(\lim \mathcal{N})]$. Since $\{a\}$ must be an extension of the empty set satisfying the heads of clauses whose bodies is satisfied in $[\emptyset, Lit(\lim \mathcal{M})]$, it is impossible that $\{a, b\}$ is a minimal such extension. Both of these minimal models start with a partial model corresponding to the empty set.

Note that formula (4) above uses an atom $\neg Cb$ on the left of the implication (corresponding to Cb in the disjunction). So the answer to the general question of translatability of minimal temporal partial logic into logic programming with stable generated models is that only a restricted part of minimal temporal partial logic can be translated back.

Proposition 5.47 states that logic programs endowed with the stable generated semantics can be faithfully translated into minimal temporal partial logic. This immediately also means that we can translate non-disjunctive extended logic programs with the answer set semantics into minimal temporal partial logic.

Corollary 5.50 (Answer sets in temporal logic) For every non-disjunctive extended logic program S the following holds:

1. For every coherent answer set I of S, there is a $\preceq^g$-minimal temporal partial model of $\rho(S)$ such that its limit corresponds to I.

2. For every $\preceq^g$-minimal temporal partial model M of $\rho(S)$, the set of literals true in lim M is a coherent answer set of S.

Proof: This is immediate from the correspondences of Propositions 5.38 and 5.47. □

It is easy to see that this result is also true for $\preceq^{gel}$-minimal models. Using the latter semantics, we can also embed logic programs with *supported* models into temporal partial logic (for more information about supported models, the other semantics, and logic programming in general, the interested reader is referred to for instance [Apt90] or [Llo87]). The translation needed is different (analogously to the difference between the translation of default logic with Reiter extensions and default logic with weak extensions into MTEL*).

Proposition 5.51 Define the translation $\overline{\rho}$ on non-disjunctive sequents without classical negation as follows

$$\overline{\rho}(q_1, \ldots, q_m, -r_1, \ldots, -r_n \Rightarrow p) =$$
$$FCq_1 \wedge \ldots \wedge FCq_m \wedge \neg FCr_1 \wedge \ldots \wedge \neg FCr_n \rightarrow XCp.$$

For a set S of such sequents, define $\overline{\rho}(S) = \{\overline{\rho}(s) \mid s \in S\}$. Then for every non-disjunctive logic program without classical negation S the following holds:

1. For every supported model I of S, there is a $\preceq^{gel}$-minimal temporal partial model of $\overline{\rho}(S)$ such that its limit corresponds to I.

2. For every $\preceq^{gel}$-minimal temporal partial model M of $\overline{\rho}(S)$, the set of atoms true in lim M is a supported model of S.

Proof: In [MT93], the following translation *dl* from logic programs of the above sort into default logic is given:

$$dl(q_1, \ldots, q_m, -r_1, \ldots, -r_n \Rightarrow p) =$$

$$\frac{q_1 \wedge \ldots \wedge q_m : \neg r_1, \ldots, \neg r_n}{p},$$

and $dl(S) = \{dl(s) \mid s \in S\}$. Supported models of S (in the traditional definition) are in a one-to-one correspondence with weak extensions of the default theory $\langle dl(S), \emptyset \rangle$. On the other hand, we have given a translation σ from default theories to TEL-formulae, such that there is a correspondence between weak extensions of a theory and $\preceq^{gel}$-minimal models of the translation. Composing these two translations yields the TEL-formula (for the above sequent):

$$FK(q_1 \wedge \ldots \wedge q_m) \wedge \neg FKr_1 \wedge \ldots \wedge \neg FKr_n \rightarrow XKp$$

Furthermore, $FK(q_1 \wedge \ldots \wedge q_m)$ is equivalent (on conservative models) to $FK(q_1) \wedge \ldots \wedge FK(q_m)$. The result then follows from a correspondence between $\preceq^{gel}$-minimal temporal epistemic models and $\preceq^{gel}$-minimal temporal partial models for formulae that use only objective formulae which are literals. □

The temporal semantics of logic programs presented here has some similarities with the approach of Lin and Reiter ([LR96]) who give a semantics of logic programs in the situation calculus. We will discuss their approach and compare it with ours in Section 5.8.

5.4 A classical proof system

In this section we will apply our approach of temporal specification of reasoning to a relatively simple type of reasoning: based on a classical proof system. We will show how proof rules can be represented by temporal formulae. As an example, consider Modus Ponens:

$$\frac{A \quad A \rightarrow B}{B}$$

Here A and B are meta-variables ranging over the set of formulae, and $A \rightarrow B$ is a term structure built from them using the logical connective $\rightarrow$. We want the temporal models to reflect the proof process, such that an information state at a certain point in time reflects what has been derived up to that moment. The temporal interpretation of such a proof rule we have in mind is the following:

> *For any formulae A and B, if in the current information state both A and $A \rightarrow B$ have been derived then in a next information state B is derived.*

This interpretation of modus ponens is formalized by the following temporal axiom scheme (for all formulae A and B):

$$C(A) \wedge C(A \rightarrow B) \rightarrow \exists \mathrm{X} C(B).$$

However, in most information state frames (certainly the ones of Chapter 2), if the formulae A and $A \rightarrow B$ are in the theory of a current state then so is B. As we want to describe the steps of reasoning by time steps this is undesirable. One solution is to take as information states just sets of formulae, not necessarily closed under propositional provability. We prefer, nevertheless, states of a more semantical nature. An option is to extend the notion of partial model to the notion of valuation of all formulae, in a manner similar to [BM92], see also [San85]. For each formula φ of the original language, we define a new atom at_φ, and then we take the propositional language induced by these new atoms as our new language. So if $\mathcal{L}_P$ denotes the set of formulae based on a set of propositional atoms P, then we define a new set of atoms $P' = \{\mathsf{at}_\varphi \mid \varphi \in \mathcal{L}_P\}$. So we have a natural bijection $\varphi \rightarrow \mathsf{at}_\varphi$ between $\mathcal{L}_P$ and P'. Notice that P is embedded in P' by $P \ni p \ \rightarrow \ \mathsf{at}_p \in P'$. We will use the branching time logic of Section 4.3 based on the information state frame IS^{3val} of partial models (see Definition 2.3) with $\mathcal{L}_{P'}$ as the language.

We can now describe any instance of the proof rule Modus Ponens by a temporal formula as follows:

$$C(\mathsf{at}_\varphi) \wedge C(\mathsf{at}_{\varphi \rightarrow \psi}) \rightarrow \exists \mathrm{X} C(\mathsf{at}_\psi).$$

This allows us to give a temporal axiomatization of a proof system. In addition we need a temporal translation of the initial axioms: the theory from which conclusions are to be drawn. Suppose K is any set of formulae of $\mathcal{L}_P$. Let $\mathsf{at}(K)$ be the set of atoms corresponding to the formulae in K. We require that these atoms are true at each moment of time. Therefore, for any such formula φ we can simply add the formula $C(\mathsf{at}_\varphi)$ to our temporal theory.

After these preparations we are ready to formalize the translation of the proof rules into temporal formulae:

Definition 5.52

1. By $\mathsf{Forterm}$ we denote the set of term structures built up from (meta-) variables, ranging over $\mathcal{L}_P$, by use of the logical connectives. A *proof system PS* is a set of *proof rules* of type $(A_1, \ldots, A_k)/B$ where the $A_i, B \in \mathsf{Forterm}$. Let a proof rule $PR : (A_1, \ldots, A_k)/B$ be given en let MV_{PR} be the set of *meta-variables* occurring in $A_1, \ldots, A_k$ and B. A mapping $\sigma : MV_{PR} \rightarrow \mathcal{L}_P$ is called a *meta-variable assignment*. Any meta-variable assignment σ can be extended in a canonical manner to a substitution mapping

$$\sigma^* : \mathsf{Forterm} \rightarrow \mathcal{L}_P$$

such that σ^* substitutes formulae for the meta-variables of MV_{PR} in any term structure of $\mathsf{Forterm}$.

The *temporal translation* of a proof rule PR of type $(A_1, \ldots, A_k)/B$ is the set T_{PR} of instances of temporal formulae defined by:

$$\{C(\mathsf{at}_{\sigma^*(A_1)}) \wedge \ldots \wedge C(\mathsf{at}_{\sigma^*(A_k)}) \rightarrow \exists \mathrm{X} C(\mathsf{at}_{\sigma^*(B)}) \mid \\ \sigma \text{ meta-variable assignment for } PR\}.$$

The *temporal translation* T_{PS} of PS is defined by: $T_{PS} = \bigcup_{PR \in PS} T_{PR}$.

2. Let K be any set of objective formulae of $\mathcal{L}_P$. The *temporal translation* T_K of K is defined by: $T_K = \{C(\mathsf{at}_\varphi) \mid \varphi \in K\}$.

3. We have to make sure that once a fact has been established, it remains known at all later points (conservativity); this can be axiomatized by the temporal theory

$$Cons = \{PC(a) \rightarrow C(a) \mid a \in P'\}.$$

The overall translation of proof rules and theory is defined by:

$$Th_{PS,K} = T_{PS} \cup T_K \cup Cons.$$

Some proof systems may consist of both proof rules and axioms; these may be incorporated by adding them to the theory K.

The temporal theory $Th_{PS,K}$ prescribes that certain conclusions have to be drawn at certain point in time. Analogously to the case of default logic in (branching time) temporal logic, we want those conclusions to be the only ones. To this end, we will consider minimal models of this theory. The ordering is the same as the one used in Section 5.2 (see Definition 5.11), but with a different information state frame: for two branching time partial models $\mathcal{M}, \mathcal{N}$, we have $\mathcal{M} \preceq^g_{br} \mathcal{N}$ if they are based on the same flow of time and for all time points s: $\mathcal{M}(s) \preceq \mathcal{N}(s)$, where $\preceq$ is the information order on IS^{3val}.

A first observation about this temporal theory $Th_{PS,K}$ is that there exist temporal partial models of it. Such a model could be constructed incrementally, starting with a root, adding its successor partial models in the next step, and any time the model has been constructed up until a certain level, one can construct the next level by adding successor partial models to those at the current level. This is possible since the formulae of $Th_{PS,K}$ prescribe existence of successors, obeying certain properties. It is easy to see that these properties are never contradictory since only truth of certain atoms is prescribed. Taking such a model and changing the truth value of atoms which are not prescribed to be true by $Th_{PS,K}$ to undefined, points out a manner to establish the existence of minimal models of $Th_{PS,K}$. In the following theorem, $K \vdash_{PS} \varphi$ denotes the fact that φ is derivable from K in the system PS, where this notion is defined as usual. We will assume that there are no proof rules with empty premise: the conclusions of such a rule can always be considered as part of the initial axioms K.

Theorem 5.53 Let PS be any proof system and K any set of formulae of $\mathcal{L}_P$. Let $\mathcal{M}$ be a minimal temporal partial model of $Th_{PS,K}$. For any formula φ of $\mathcal{L}_P$ it holds

$$K \vdash_{PS} \varphi \Leftrightarrow \mathcal{M} \models \neg P(\top) \rightarrow \exists \mathrm{F} C(\mathsf{at}_\varphi).$$

Proof: "$\Rightarrow$" Suppose $K \vdash_{PS} \varphi$ and suppose that $\psi_1, \ldots, \psi_{n-1}, \psi_n$, with $\psi_n = \varphi$, is a proof for φ. For a non-minimal element $t \in T$ (we assume $\mathcal{M}$ is based on $(T, <)$) it holds trivially that $(\mathcal{M}, t) \models \neg P(\top) \rightarrow \exists \mathrm{F} C(\mathsf{at}_\varphi)$, so let r be a minimal element in T. We shall prove the following by induction:
For every $1 \leq i \leq n$ there is a time point s reachable from r such that $\mathsf{at}_{\psi_1}, \ldots, \mathsf{at}_{\psi_i}$ are true in $\mathcal{M}$ at time point s.
$i = 1$: ψ_1 has to be an element of K (remember that we assumed that rules never have an empty premise) and as $\mathcal{M}$ is a model of T_K, at_φ has to be true in $\mathcal{M}$ at time point r.
$i \rightarrow i+1$: suppose that s is a time point reachable from r and that $\mathsf{at}_{\psi_1}, \ldots, \mathsf{at}_{\psi_i}$ are true in $\mathcal{M}$ at time point s. If ψ_{i+1} is an element of K then the same argument as above yields that $\mathsf{at}_{\psi_{i+1}}$ must be true in $\mathcal{M}$ at point s, so assume that ψ_{i+1} is the result of applying a proof rule PR to a subset of the formulae $\psi_1, \ldots, \psi_i$ (say $\alpha_1, \ldots, \alpha_k$). Then there is a rule $C(\mathsf{at}_{\alpha_1}) \wedge \ldots \wedge C(\mathsf{at}_{\alpha_k}) \rightarrow \exists \mathrm{X} C(\mathsf{at}_{\psi_{i+1}})$ in T_{PS} which has to be true in $\mathcal{M}$ at point s. As $\mathsf{at}_{\alpha_1}, \ldots, \mathsf{at}_{\alpha_k}$ are true in $\mathcal{M}$ at point s, there has to be a successor t to s in which $\mathsf{at}_{\psi_{i+1}}$ is true. The rules in *Cons* ensure that $\mathsf{at}_{\psi_1}, \ldots, \mathsf{at}_{\psi_i}$ have to be true in $\mathcal{M}$ at point t too.

Taking n for i we have that there must be a point s reachable from r such that at_{ψ_n} is true in $\mathcal{M}$ at point s. It follows that $(\mathcal{M}, s) \models \neg P(\top) \rightarrow \exists \mathrm{F} C(\mathsf{at}_\varphi)$.
"$\Leftarrow$" Suppose there is a formula φ and a minimal element r such that $(\mathcal{M}, r) \models \exists \mathrm{F} C(\mathsf{at}_\varphi)$ although $K \nvdash_{PS} \varphi$. Take the formula φ at minimal depth, i.e., if s is a point at minimal depth for which $(\mathcal{M}, s) \models \mathsf{at}_\varphi$, then there is no formula α such that there is a point t at smaller depth than s with $(\mathcal{M}, t) \models \mathsf{at}_\alpha$ but $K \nvdash_{PS} \alpha$. As $\mathcal{M}$ is a minimal model of $Th_{PS,K}$, if at_φ were undefined in $\mathcal{M}$ at point s, a formula in $Th_{PS,K}$ would become false. If this is a formula from T_K then it has to be the formula $C(\mathsf{at}_\varphi)$, but then φ is in K and therefore $K \vdash_{PS} \varphi$. If it is a formula in *Cons* then it must be the rule $PC(\mathsf{at}_\varphi) \rightarrow C(\mathsf{at}_\varphi)$ at time point s. This means that at_φ is true in a point at smaller depth, which was not the case. Therefore, it must be a rule of T_{PS}, say $C(\mathsf{at}_{\alpha_1}) \wedge \ldots \wedge C(\mathsf{at}_{\alpha_k}) \rightarrow \exists \mathrm{X} C(\mathsf{at}_\varphi)$ which will become false in a point t with $t < s$. But as $\mathsf{at}_{\alpha_1}, \ldots, \mathsf{at}_{\alpha_k}$ have to be true in $\mathcal{M}$ at point t and t is at smaller depth than s, we must have that $K \vdash_{PS} \alpha_1, \ldots, K \vdash_{PS} \alpha_k$. But there is a proof rule in PS which can be applied to $\alpha_1, \ldots, \alpha_k$ yielding φ, and therefore $K \vdash_{PS} \varphi$. This shows that such a formula can not exist. $\Box$

Note that for a minimal temporal partial model of $Th_{PS,K}$ the partial models of time points which are minimal elements (according to Definition 4.16), are the same (atoms corresponding to formulae of K are true, other atoms are undefined). In this way a semantics is defined which can be seen as a generalization of the manner in which modal and temporal semantics can be given to intuitionistic logic (see [Gab82], [Kri65]). This approach can be used for any proof system.

The rules in $Th_{PS,K}$ only prescribe *truth* of atoms of P', never truth of a negation. Minimal models of this theory therefore never assign false to any atom. This corresponds to the fact that a proof system (in the classical sense) only establishes validity of formulae. It makes the use of partial models seem unnecessary: the information state frame IS^{2val} of two-valued states (Definition 2.2) could have been used. But systems have also been defined to 'prove' non-validity (see for example [Cai78] for a Hilbert-style system). An example of such a 'non-proof' rule would be that if A can not be proved, then $A \wedge B$ can not be proved. We would be able to formulate this as

$$C(\neg\mathsf{at}_A) \rightarrow \exists \mathrm{X} C(\neg\mathsf{at}_{A\wedge B}).$$

The class of branching time models is large: a branching time model consists of any set, with an ordering that defines a forest, and a labeling function. There are many ways in which this class can be reduced, for instance by placing constraints on the underlying set T of time points, and/or by constraining the cardinality of minimal elements and successors. Let us now suppose that the class of branching time models has been constrained to such a degree that it forms a set. Then we have the following.

Proposition 5.54 The set of models of the temporal theory $Th_{PS,K}$ has a final model $F_{PS,K}$.

Proof: The theory $Th_{PS,K}$ consists of formulae which are forward persistent under any homomorphism (this easily follows from Theorem 4.26). Then Theorem 4.41 can be applied. □

If we have a proof $\varphi_1, \dots, \varphi_n$ of which (only) the first k formulae are axioms from K, then a *proof trace* is a sequence $(\mathcal{M}_i)_{i=0\dots n-k}$ of partial models such that $Lit(\mathcal{M}_i) = \{\mathsf{at}_{\varphi_j} \mid j = 1, \dots, k+i\}$. In such a trace the partial model $\mathcal{M}_i$ reflects exactly the formulae which have been derived up until the i^{th} step of the proof. It is easy to see that, although such a proof trace itself is in general not a model of $Th_{PS,K}$, it can always be embedded in the final model $F_{PS,K}$. Note that for a branch $\mathcal{B}$ of a minimal model, the limit model $\lim_{\mathcal{B}} F_{PS,K}$ corresponds to the set of all conclusions drawn in that reasoning pattern; this is a subset of the deductive closure of K under PS (since we allow non-exhaustive reasoning patterns).

These proof sequences correspond to the traces assigned to the axioms by the reasoning frame operator we can associate to $Th_{PS,K}$, in a manner similar to what was done for the branching time case of default logic.

$$\begin{aligned}\mathcal{T}_{Th_{PS,K}}(X) = \{\mathcal{B} \mid\ & \text{there is a temporal partial model } \mathcal{M} \text{ such that} \\ & \mathcal{B} \in \mathsf{Br}(\mathcal{M}), \mathcal{M} \models_{\preceq^g_{br}} Th_{PS,K} \text{ and } Th(\mathcal{B}_0) = Cn(X)\}.\end{aligned}$$

5.5 GK

The logic GK of knowledge and justified assumptions[1] was introduced by Lin and Shoham in [LS92]. The reason we consider this logic here, is that both default logic and autoepistemic logic can be embedded in it, hence it can claim, to a certain degree, to be a general logic of defeasible reasoning. Also, a correspondence result between this logic and MTEL* will yield an embedding of autoepistemic logic into MTEL*. We will first give a brief account of GK; this will be based on (but not exactly equal to) the presentation of [ST94] rather than directly on [LS92], since the former is, in our opinion, more intuitive, and also makes the connection with MTEL* easier.

The language of GK is a propositional modal language, with two modal operators, K (for *knowledge*) and A (for *assumptions*). The formula $A\varphi$ means that the agent has *assumed* φ. A GK-model is a triple (w, M_K, M_A), where w is a propositional valuation (the *actual* world), and M_K and M_A are sets of propositional valuations such that $M_A \subseteq M_K$. The truth of a formula in a model is defined inductively, where for a propositional formula φ we have $(w, M_K, M_A) \models_{GK} \varphi$ if $w \models \varphi$ (in propositional logic). The boolean cases are standard, and for the modal operators the semantical clauses are as follows:

$$\begin{array}{lcl}(w, M_K, M_A) \models_{GK} K\varphi & \Leftrightarrow & (v, M_K, M_A) \models_{GK} \varphi \text{ for all } v \in M_K \\ (w, M_K, M_A) \models_{GK} A\varphi & \Leftrightarrow & (v, M_K, M_A) \models_{GK} \varphi \text{ for all } v \in M_A.\end{array}$$

The requirement on models that $M_A \subseteq M_K$ corresponds to the validity of $K\varphi \to A\varphi$ for propositional φ: everything known is also assumed.

Lin and Shoham introduce a preference ordering on GK-models which favors models with less knowledge, under equal assumptions. For a model $\mathsf{M} = (w, M_K, M_A)$, let $K(\mathsf{M}) = \{\alpha \mid \alpha$ propositional and $\mathsf{M} \models_{GK} K\alpha\}$ and $A(\mathsf{M}) = \{\alpha \mid \alpha$ propositional and $\mathsf{M} \models_{GK} A\alpha\}$. If $\mathsf{M} = (w, M_K, M_A)$ and $\mathsf{N} = (v, N_K, N_A)$ then M is *GK-preferred* over N if

- $K(\mathsf{M}) \subset K(\mathsf{N})$, and
- $A(\mathsf{M}) = A(\mathsf{N})$.

where $\subset$ denotes strict inclusion. For a set of formulae S, the model M is a *minimal* model of S if it is a model of S and there is no model of S GK-preferred over M. The minimal models are not yet the intended models; an extra criterion is imposed. A model M is a *preferred* model of S if it is a minimal model of S *and* $K(\mathsf{M}) = A(\mathsf{M})$. This corresponds to the intuition that the assumptions must be justified, or grounded, (they must be known) in a preferred model. Finally, semantic entailment is defined by

$$S \models_{GK} \varphi \;\Leftrightarrow\; \varphi \text{ holds in all preferred models of } S.$$

One of the key results in [LS92] is the following characterization result for preferred models similar to our results in Section 4.5:

[1]The letters GK stand for 'Grounded Knowledge'.

Theorem 5.55 (Lin and Shoham) Let S be a set of formulae of the form

$$K\alpha \wedge A\beta \wedge \neg A\gamma_1 \wedge \ldots \wedge \neg A\gamma_n \rightarrow K\varphi$$

where $\alpha, \beta, \gamma_1, \ldots, \gamma_n$ and φ are propositional formulae, $n \geq 0$, and both $K\alpha$ and $A\beta$ may be absent. A model M is a preferred model of S if and only if:

1. $K(\mathsf{M}) = A(\mathsf{M})$, and
2. $K(\mathsf{M}) = \bigcup_{i=0}^{\infty} E_i$ where the E_i are defined as follows:
   (a) $E_0 = Cn(\{\varphi \mid K\varphi \in S\})$, and for $i \geq 0$:
   (b) $E_{i+1} = Cn(E_i \cup \{\varphi \mid$ there is a formula $K\alpha \wedge A\beta \wedge \neg A\gamma_1 \wedge \ldots \wedge \neg A\gamma_n \rightarrow K\varphi \in S$ such that $\alpha \in E_i$, $\beta \in A(\mathsf{M})$, and $\gamma_1, \ldots, \gamma_n \notin A(\mathsf{M})\})$.

This theorem is of course very similar to the characterization results for minimal models in Section 4.5.

The key idea behind the correspondence between GK and MTEL* is that a formula $A\varphi$ corresponds to the formula $FK\varphi$: in temporal logic, a (justified) assumption is a formula that has to be derived *at some point in time*, but not necessarily before it is used in the derivation of another formula. So consider the following translation Ω of formulae of the form described in Theorem 5.55 to TEL-formulae:

$$\Omega(K\varphi) = K\varphi, \text{ and}$$

$$\Omega(K\alpha \wedge A\beta \wedge \neg A\gamma_1 \wedge \ldots \wedge \neg A\gamma_n \rightarrow K\varphi) = K\alpha \wedge FK\beta \wedge \neg FK\gamma_1 \wedge \ldots \wedge \neg FK\gamma_n \rightarrow XK\varphi.$$

This is a faithful embedding, as the next theorem shows.

Theorem 5.56 Let S be a set of formulae of the form described in Theorem 5.55. Then

1. if M is a preferred model of S, then there exists a $\preceq^{gel}$-minimal TELC-model $\mathcal{N}$ of $\Omega(S)$ with $Th(\lim \mathcal{N}) = K(\mathsf{M})$.
2. if $\mathcal{M}$ is a $\preceq^{gel}$-minimal TELC-model of $\Omega(S)$, then the GK-model $\mathsf{N} = (w, \lim \mathcal{M}, \lim \mathcal{M})$, where w is any propositional valuation, is a preferred model of S.

Proof: We need the following characterization result for $\preceq^{gel}$-minimal models. A TELC-model $\mathcal{M}$ is a $\preceq^{gel}$-minimal model of a theory S consisting of formulae of the form $K\varphi$ and $K\alpha \wedge FK\beta \wedge \neg FK\gamma_1 \wedge \ldots \wedge \neg FK\gamma_n \rightarrow XK\varphi$ if and only if

$$\begin{array}{lcl} \mathcal{M}_0 & = & Mod(\{\varphi \mid K\varphi \in S\}), \text{ and for } i \geq 0: \\ \mathcal{M}_{i+1} & = & Mod(Th(\mathcal{M}_i) \cup \{\varphi \mid \text{ there is a formula} \\ & & K\alpha \wedge FK\beta \wedge \neg FK\gamma_1 \wedge \ldots \wedge \neg FK\gamma_n \rightarrow XK\varphi \in S \text{ with} \\ & & \mathcal{M}_i \models K\alpha, \lim \mathcal{M} \models K\beta, \text{ and } \lim \mathcal{M} \not\models K\gamma_1, \ldots, K\gamma_n\}). \end{array}$$

This result is not proved in Section 4.5, but it easily follows from Proposition 4.58 using similar arguments as in the proof of Corollary 4.55 (which is almost the above characterization result, but for MTEL). With this characterization and Theorem 5.55 the proof is straightforward. □

The theory $\Omega(S)$ again specifies a reasoning frame operator that formalizes reasoning with GK, according to Definition 4.63:

$$\mathcal{T}_{\Omega(S)}(X) = \{\mathcal{M} \mid \mathcal{M} \models_{\preceq^{gel}} \Omega(S) \text{ and } Th(\mathcal{M}_0) = Cn(X)\}.$$

Lin and Shoham give a faithful translation of default theories into GK (where preferred models of the translation correspond to extensions). The composition of their translation and the above translation Ω is equal to our translation τ of default theories into TEL-formulae, which was faithful with respect to MTEL*. In [LS92], also a faithful translation from autoepistemic logic (see [Moo85]) into GK is given. This immediately gives us a translation of autoepistemic theories into TEL-formulae. Alternatively, there is a translation of autoepistemic logic into default theories, which is faithful if weak extensions are used (see [MT93]). Then we can use our translation of default logic with weak extensions into MTEL*. The composition of these two translations would yield the same result as the route we will be taking. We will first briefly introduce autoepistemic logic.

5.6 Autoepistemic logic

The language of autoepistemic logic is a propositional modal language with a modal operator L for belief; this language is denoted by $\mathcal{L}_L$. First we need the notion of a stable set (originally proposed in [Sta93]; see also [MT93]).

Definition 5.57 A set of sentences $S \subseteq \mathcal{L}_L$ is a *stable* set if

1. S is closed under propositional consequence.
2. For any $\varphi \in \mathcal{L}_L$, if $\varphi \in S$ then $L\varphi \in S$.
3. For any $\varphi \in \mathcal{L}_L$, if $\varphi \notin S$ then $\neg L\varphi \in S$.

For a stable set S, the set of purely propositional formulae in S, called the *kernel* of S, is denoted as $\ker(S)$.

For every set of propositional formulae T, there is a unique stable set S such that $T = \ker(S)$ (see [MT93]).

Now let I be a set of sentences in $\mathcal{L}_L$. A set of $\mathcal{L}_L$-formulae S is a *stable expansion* of I, if

$$S = Cn(I \cup \{L\varphi \mid \varphi \in S\} \cup \{\neg L\varphi \mid \varphi \notin S\})$$

where Cn is propositional consequence. It was proved in [Kon88b] that any set of $\mathcal{L}_L$ formulae can be converted into a set with the same stable expansions, whose formulae are in the normal form:

$$L\alpha \wedge \neg L\beta_1 \wedge \ldots \wedge \neg L\beta_n \rightarrow \gamma$$

where $\alpha, \beta_1, \ldots, \beta_n$ and γ are propositional. This means that we can assume, without loss of generality, that $\mathcal{L}_L$ theories are in normal form. For such theories, the following translation is defined in [LS92]:

$$L\alpha \wedge \neg L\beta_1 \wedge \ldots \wedge \neg L\beta_n \rightarrow \gamma$$
$$\Longrightarrow$$
$$A\alpha \wedge \neg A\beta_1 \wedge \ldots \wedge \neg A\beta_n \rightarrow K\gamma.$$

It is proved in [LS92] that a consistent stable set S of $\mathcal{L}_L$ formulae is a stable expansion of a set I of $\mathcal{L}_L$-formulae in normal form if and only if there is a preferred model M of the translation of I such that $K(\mathsf{M}) = \ker(S)$.

Now we can use our translation Ω (defined just above Theorem 5.56) to arrive at an embedding of autoepistemic logic in MTEL*.

Theorem 5.58 Let I be a set of $\mathcal{L}_L$-formulae in normal form. Define I' as

$$I' = \{K\varphi \mid \varphi \in I, \text{ for propositional } \varphi\} \cup$$
$$\{FK\alpha \wedge \neg FK\beta_1 \wedge \ldots \wedge \neg FK\beta_n \rightarrow XK\gamma \mid L\alpha \wedge \neg L\beta_1 \wedge \ldots \wedge \neg L\beta_n \rightarrow \gamma \in I\}.$$

A consistent stable set of $\mathcal{L}_L$ formulae S is a stable expansion of I if and only if there is a $\preceq^{gel}$-minimal model $\mathcal{M}$ of I' such that $Th(\lim\ \mathcal{M}) = \ker(S)$.

Proof: The translation is the composition of Ω and the translation of autoepistemic logic into GK. The statement of the theorem follows from the faithfulness of both. Alternatively, the translation of autoepistemic logic into default logic (with weak extensions, see [MT93]) can be composed with our translation of default logic (with weak extensions) into MTEL* (see Proposition 5.9), to give the same correspondence. $\square$

The theory I' induces the reasoning frame operator associated with autoepistemic reasoning with the autoepistemic theory I, according to Definition 4.63:

$$\mathcal{T}_{I'}(X) = \{\mathcal{M} \mid \mathcal{M} \models_{\preceq^{gel}} I' \text{ and } Th(\mathcal{M}_0) = Cn(X)\}.$$

The different embeddings for default logic and autoepistemic logic give a temporal view of their difference ([LS92] gives a knowledge/assumption view). The correspondence for default logic is

$$(\alpha : \beta_1, \ldots, \beta_n)/\gamma$$
$$\Longleftrightarrow$$
$$K\alpha \wedge \neg FK\neg\beta_1 \wedge \ldots \wedge \neg FK\neg\beta_n \rightarrow XK\gamma$$

and the autoepistemic correspondence is

$$L\alpha \wedge \neg L\neg\beta_1, \ldots, \neg L\neg\beta_n \rightarrow \gamma$$
$$\Longleftrightarrow$$
$$FK\alpha \wedge \neg FK\neg\beta_1 \wedge \ldots \wedge \neg FK\neg\beta_n \rightarrow XK\gamma.$$

This means that the difference lies in the temporal interpretation of the premise α: in default logic, it has to be known (derived) *now* in order to be used, whereas in autoepistemic logic, it is sufficient if it is derived *sometimes* (in the future).

The MTEL* input and reasoning formulae are thus a generalization of both default logic and autoepistemic logic rules. Mixing of these two is allowed, and disjunctive conclusions are allowed. In Section 5.3, it was shown that disjunctive logic programs (with stable generated semantics) can be faithfully translated into MTEL*. However, it will turn out that the above translation of GK into MTEL*, when extended in the obvious way to handle disjunctive conclusions, is no longer faithful. The next section treats disjunctive rules in logic programming, GK, and default logic.

5.7 Disjunctive rules

In Section 5.3 it was shown that logic programs with the stable generated semantics can be faithfully translated into temporal partial logic with $\preceq^g$-minimal models. It is straightforward to show that we can also take $\preceq^{gel}$-minimal models. For extended logic programs (whose rules do not contain disjunction in the head) the stable generated models coincide with answer sets ([GL91]). For disjunctive logic programs example 5.39 shows that stable generated models do not coincide with answer sets. This also means that the translation of logic programs into temporal partial logic does not in general preserve the answer set semantics.

There is also a variant of default logic which allows disjunctive conclusions. Of course, a default rule may have a disjunction in the conclusion. If the conclusion is for example $p \vee q$, then the formula $p \vee q$ will be in an extension if the rule was applicable. What disjunctive conclusions formalize, is that a commitment to one of the disjuncts should be made. A conclusion $p|q$ (where the symbol '|' separates the disjunctive conclusions) means that when the rule is applicable, either the formula p or the formula q should be concluded, not only the formula $p \vee q$. We will briefly describe disjunctive default logic, as introduced in [GLPT91].

A disjunctive default is an expression of the form $(\alpha : \beta_1, \ldots, \beta_n)/\gamma_1| \ldots | \gamma_n$, where α, β_i and γ_j are propositional formulae.

Definition 5.59 ([GLPT91]) Let D be a set of disjunctive defaults and E a set of sentences. $Min_E(D)$ is the set of all minimal deductively closed sets M satisfying the following condition for every $(\alpha : \beta_1, \ldots, \beta_m)/\gamma_1| \ldots |\gamma_n \in D$: if $\alpha \in M$, and $\neg\beta_1, \ldots, \neg\beta_m \notin E$, then $\{\gamma_1, \ldots, \gamma_n\} \cap M \neq \emptyset$. E is an extension of D if $E \in Min_E(D)$.

Note that there is no mention of a set W of axioms. This is not a severe limitation since an axiom φ can be written as a default rule $(\ :\ )/\varphi$ which is always applicable (this presupposes that a default rule may have an empty set of justifications!). Alternatively, the definition can easily be adapted to include a set W of axioms. The definition clearly generalizes the original fixed point definition (Proposition 3.4) of an extension. The function τ translating default theories into TEL can be extended to handle disjunctive rules in a straightforward manner.

Definition 5.60 Define the mapping τ from defaults to TEL-formulae by

$$\tau : (\alpha : \beta_1, \ldots, \beta_m)/\gamma_1|\ldots|\gamma_n \mapsto$$
$$K\alpha \wedge \neg FK(\neg\beta_1) \wedge \ldots \wedge \neg FK(\neg\beta_m) \rightarrow XK\gamma_1 \vee \ldots \vee XK\gamma_n.$$

The $\preceq^g$-minimal models (or the $\preceq^{gel}$-minimal models) of the τ translation of a default theory do not necessarily correspond to its extensions. This is not surprising, since disjunctive default logic has the same behavior as answer sets. If a program rule $a_1, \ldots, a_n, -b_1, \ldots, -b_m \Rightarrow c_1, \ldots, c_k$ (where the a_i, b_i and c_i are classical literals) is translated into the default rule

$$\frac{a_1 \wedge \ldots \wedge a_n : \neg b_1, \ldots, \neg b_m}{c_1|\ldots|c_k}$$

then there is a one-to-one correspondence between answer sets of the program and extensions of the translation (see [GLPT91]).

Of course, the temporal interpretation of default logic was inspired by the other definition of extension (Definition 3.1, with the sets E_i), which is equivalent to the fixed point definition. However, it turns out that if we generalize the former definition (Definition 3.1) to the disjunctive case, we get a notion of extension which is not equivalent to the above definition (Definition 5.59).

Definition 5.61 Let D be a set of disjunctive defaults. A deductively closed set E of sentences is a *generated extension* of D if there is a sequence $E_0 \subseteq E_1 \subseteq \ldots$ of deductively closed sets of sentences such that

1. $E_0 = Cn(\emptyset)$;
2. E_{n+1} is a minimal extension of E_n satisfying the following closure condition: if $(\alpha : \beta_1, \ldots, \beta_k)/\gamma_1|\ldots|\gamma_l \in D$, $\alpha \in E_n$, and $\neg\beta_1, \ldots, \neg\beta_k \notin E$ then $\{\gamma_1, \ldots, \gamma_l\} \cap E_{n+1} \neq \emptyset$.

and $E = \bigcup_{n=0}^{\infty} E_n$.

It is easy to see that this definition coincides with Definition 3.1 of an extension if D consists of non-disjunctive rules. We can use the default translation of the example of Section 5.3 to show that the two definitions of extension and generated extension are different.

Example 5.62 Let $D = \{(:)/a|b, (a :)/b, (: \neg a)/a\}$. Then $E = Cn(\{a, b\})$ is a generated extension of D: $E_0 = Cn(\emptyset)$, $E_1 = Cn(\{a\})$, $E_2 = Cn(\{a, b\})$, $E_i = E_2$ for $i > 2$. But E is not an extension. It does satisfy the closure condition with respect to itself, but $Cn(\{b\})$ also satisfies it, so $Min_E(D) = \{Cn(\{b\})\}$.

The translation,

$$\begin{array}{l} XKa \vee XKa \\ Ka \rightarrow XKb \\ \neg FKa \rightarrow XKa \end{array}$$

has one $\preceq^g$-minimal model, corresponding to the generated extension. This is not a coincidence, as the following theorem shows.

Theorem 5.63 Let D be a set of disjunctive defaults, and $\tau(D)$ its translation into TEL-formulae. Then a set of sentences E is a generated extension if and only if it is equal to $Th(\lim \mathcal{M})$ for some $\preceq^g$-minimal model of $\tau(D)$. Moreover, this correspondence between a generated extension E of D and a $\preceq^g$-minimal model $\mathcal{M}$ of $\tau(D)$ can be chosen such that $E_i = Th(\mathcal{M}_i)$, where the sets E_i are as in Definition 5.61. The correspondence also holds when taking $\preceq^{gel}$-minimal models of $\tau(D)$.

Proof: The proof is completely analogous to the proofs of Propositions 5.3 and 5.4 for the non-disjunctive case, using Proposition 4.54 directly, instead of its Corollary 4.55. $\square$

The obvious generalization of the translation of default theories into GK to the disjunctive case, leads to a correspondence between preferred models and extensions (not generated extensions). For the non-disjunctive case, we have a quite homogeneous picture: logic programs with stable generated models are equivalent to logic programs with answer sets, which can be translated into default logic, which in turn can be translated into GK and MTEL, all preserving semantics. It seems that there are at least two different approaches to the generalization of rules under these semantics to the case of disjunctive rules. The first approach is the 'static' approach of GK, answer sets, and extensions in the sense of [GLPT91], and the other one is the 'generated' approach of stable generated models, generated extensions, and MTEL.

The question then arises whether we can also specify the extension / answer set / preferred model semantics for disjunctive rules in MTEL. The answer is yes, by eliminating the temporal nature of the translation. All reasoning should be done instantly. The new translation of default logic then takes a default $(\alpha : \beta_1, \ldots, \beta_n)/\gamma_1| \ldots |\gamma_m$

to the formula

$$K\alpha \wedge \neg FK(\neg\beta_1) \wedge \ldots \wedge \neg FK(\neg\beta_m) \rightarrow K\gamma_1 \vee \ldots \vee K\gamma_n.$$

Notice that the only difference with τ is that there is no X-operator in front of the conclusions. Now we of course have to prove that this translation is faithful, i.e., that the limits of $\preceq^g$-minimal models of the translation of a default theory are extensions of this theory. Rather than doing that, we will give a general embedding of GK-formulae in MTEL*.

The general idea is (like before) that assumptions are beliefs to be deduced later, that is, a formula $A\varphi$ should be translated by $FK\varphi$. There are a number of things to remark here. First of all, apart from the question whether they are desirable, nested modalities in GK are not necessary: a formula with nested modalities can always be transformed into a formula without them (using equivalences like $AK\alpha \equiv K\alpha$). Next, we already remarked that all deduction should be done at one point in time, and we can choose the first point. This means that all formulae (about knowledge) should be true only at the first point in time. If we introduce the notation at_0 as an abbreviation for $H\bot$ then we should only use formulae of the form $at_0 \rightarrow \alpha$ (note that the formula at_0 is true only in the first point in time). We are now ready to define the translation. Call a GK-formula subjective if every atom is in the scope of a modal operator.

Definition 5.64 Define the mapping Ω' from subjective GK-formulae without nested modalities to TEL-formulae by

$$\Omega' : \varphi \;\mapsto\; (at_0 \rightarrow (\varphi[A/FK]))$$

where $\varphi[A/FK]$ denotes the result of substituting the operators FK for A throughout φ.

A GK-model corresponds to a temporal epistemic model if its knowledge is equal to what is known at time point 0 in the temporal model, and its assumptions are equal to those things becoming known at some point in time.

Lemma 5.65 Suppose we have a GK-model M and a closed TEL-model $\mathcal{N}$ such that $K(\mathsf{M}) = Th(\mathcal{N}_0)$ and $A(\mathsf{M}) = Th(\lim \mathcal{N})$. Then for every subjective GK-formula φ without nested modalities it holds:

$$\mathsf{M} \models_{GK} \varphi \Leftrightarrow \mathcal{N} \models \Omega'(\varphi).$$

Proof: For a formula of the form $K\alpha$ (with α propositional), we have $\mathsf{M} \models_{GK} K\alpha \Leftrightarrow \alpha \in K(\mathsf{M}) \Leftrightarrow \mathcal{N}_0 \models K\alpha \Leftrightarrow (\mathcal{N}, 0) \models K\alpha$. For a formula of the form $A\alpha$ (with α propositional) we have $\mathsf{M} \models_{GK} A\alpha \Leftrightarrow \alpha \in A(\mathsf{M}) \Leftrightarrow \lim \mathcal{N} \models K\alpha \Leftrightarrow (\mathcal{N}, 0) \models FK\alpha$ (using Proposition 4.9). Every subjective GK-formula without nested modalities is a boolean combination of such formulae $K\alpha$ and $FK\alpha$, so we have

$$\mathsf{M} \models_{GK} \varphi \Leftrightarrow (\mathcal{N}, 0) \models \varphi[A/FK].$$

But $(\mathcal{N}, 0) \models \psi \Leftrightarrow \mathcal{N} \models at_0 \rightarrow \psi$ for any TEL-formula ψ, so

$$\mathsf{M} \models_{GK} \varphi \Leftrightarrow \mathcal{N} \models \Omega'(\varphi)$$

as required. □

Using this lemma we can prove that there is a correspondence between preferred models of a GK-formula and $\preceq^{gel}$-minimal models of its Ω'-translation.

Theorem 5.66 Let S be a set of subjective GK-formulae without nested modalities, and $\Omega'(S)$ its translation into TEL.

1. If M is a preferred model of S, then the TELC-model $\mathcal{N}$ defined by
$$\mathcal{N}_i = \mathit{Mod}(K(\mathsf{M})) \text{ for } i \geq 0$$
is a $\preceq^{gel}$-minimal model of $\Omega'(S)$.

2. If $\mathcal{N}$ is a $\preceq^{gel}$-minimal model of $\Omega'(S)$ then the GK-model $\mathsf{M} = (w, \mathcal{N}_0, \mathcal{N}_0)$, where w is any valuation, is a preferred model of S.

Proof: We will prove the two directions:

1. Suppose M is a preferred model of S. Define $\mathcal{N}$ by $\mathcal{N}_i = \mathit{Mod}(K(\mathsf{M}))$ for $i \geq 0$. Then M and $\mathcal{N}$ satisfy the assumption of Lemma 5.65, which means that $\mathcal{M} \models \Omega'(S)$. Now suppose it is not $\preceq^{gel}$-minimal, that is, there exists a model $\mathcal{N}'$ with $\mathcal{N}' \prec^{gel} \mathcal{N}$ and $\mathcal{N}' \models \Omega'(S)$. Let M' be defined as $(w, \mathcal{N}'_0, \lim \mathcal{N}')$, where w is any valuation. Then $\mathcal{N}'$ and M' again satisfy the requirements of Lemma 5.65, so $\mathsf{M}' \models_{GK} S$. As $\mathcal{N}' \prec^{gel} \mathcal{N}$ and $\mathcal{N}$ is constant, it must be the case that $\mathcal{N}'_0 \prec \mathcal{N}_0$. This means that $K(\mathsf{M}') \subset K(\mathsf{M})$ and $A(\mathsf{M}') = A(\mathsf{M})$ which contradicts the fact that M is a preferred, and thus minimal, model of S. Therefore, $\mathcal{N}$ is a $\preceq^{gel}$-minimal model of $\Omega'(S)$.

2. Suppose $\mathcal{N}$ is a $\preceq^{gel}$-minimal model of $\Omega'(S)$, and define $\mathsf{M} = (w, \mathcal{N}_0, \mathcal{N}_0)$, where w is any valuation. First of all, $\mathcal{N}$ must be constant, that is, $\mathcal{N}_i = \mathcal{N}_0$ for all $i > 0$. For suppose not, then $\mathcal{N}_k \prec \mathcal{N}_{k+1}$ for some index k. Consider the model $\mathcal{N}'$ defined by:
$$\begin{array}{lcll} \mathcal{N}'_i & = & \mathcal{N}_i & \text{for } i \leq k \\ \mathcal{N}'_{k+1} & = & \mathcal{N}_k & \\ \mathcal{N}'_i & = & \mathcal{N}_{i-1} & \text{for } i > k+1. \end{array}$$
This just means that we have duplicated the state at time point k. It is easy to check that $\mathcal{N}'$ is a conservative model, and as $\mathcal{N}$ is conservative, one can

prove that $\mathcal{N}' \prec^{gel} \mathcal{N}$. Furthermore, $\mathcal{N}'_0 = \mathcal{N}_0$ and $\lim \mathcal{N}' = \lim \mathcal{N}$, so for any propositional formula α, we have

$$\begin{array}{lcl}(\mathcal{N}', 0) \models K\alpha & \Leftrightarrow & (\mathcal{N}', 0) \models K\alpha \text{ and} \\ (\mathcal{N}', 0) \models FK\alpha & \Leftrightarrow & (\mathcal{N}', 0) \models FK\alpha.\end{array}$$

But this means that $\mathcal{N}' \models \Omega'(S)$, which contradicts the fact that $\mathcal{N}$ was a $\preceq^{gel}$-minimal model of $\Omega'(S)$. Thus, $\mathcal{N}$ is constant. This means that $\mathcal{N}_0 = \lim \mathcal{N}$, so $\mathcal{N}$ and M satisfy the requirement of Lemma 5.65. This gives us that $\mathsf{M} \models_{GK} S$. From the definition, we have $K(\mathsf{M}) = A(\mathsf{M})$, so we only have to check that M is a minimal model of S. So suppose it is not, suppose we have a model M' which is GK-preferred to M and a model of S. Then $K(\mathsf{M}') \subset K(\mathsf{M})$ and $A(\mathsf{M}') = A(\mathsf{M})$. Define a TELC-model $\mathcal{N}'$ by setting $\mathcal{N}'_0 = Mod(K(\mathsf{M}'))$ and $\mathcal{N}'_i = \mathcal{N}_i$ for $i > 0$. Then obviously $\mathcal{N}' \prec^{gel} \mathcal{N}$. Furthermore, $\mathcal{N}'$ and M' satisfy the requirement of Lemma 5.65, so $\mathcal{N}' \models \Omega'(S)$, contradicting the fact that $\mathcal{N}$ was a $\preceq^{gel}$-minimal model of $\Omega'(S)$.

□

The theorem means that under the embedding Ω', MTEL* behaves the same as GK on non-nested subjective formulae. In particular, they behave the same on the general disjunctive GK-rules (generalizing the disjunctive variants of both default logic and autoepistemic logic) of the form

$$K\alpha \wedge A\beta \wedge \neg A\gamma_1 \wedge \ldots \wedge \neg A\gamma_n \rightarrow K\varphi_1 \vee \ldots \vee K\varphi_m.$$

The difference between the translations Ω and Ω' clearly shows that the two different interpretations of disjunctive conclusions are caused by interpreting them in an essentially temporal or essentially non-temporal way.

5.8 Conclusions and related work

The results of this chapter show that temporal logic of information, in its various forms (using some kind of minimization; linear or branching; epistemic or partial), is well-suited for specifying a wide range of reasoning processes. The main use of these temporal specifications is to be able to formally reason about the reasoning process of an agent. Properties of the reasoning process may be of interest for verification purposes. Another possible use is to establish equivalence of two descriptions of reasoning (for instance, to detect equivalence of two default theories).

When the specification of the reasoning is in a temporal logic with preferential entailment, then the reasoning (about the reasoning process) may be done either in the preferential temporal logic directly, but also in the underlying (monotonic) temporal logic. The advantage of using the underlying logic is that it may be computationally less complex, and that it may have an axiomatic system which the preferential logic might lack. On the other hand, it might be insufficiently strong to derive all (relevant) properties of the reasoning process. In Section 9.1 some of these issues are discussed for MTEL and TEL.

Another advantage of temporal specification of reasoning is that it can provide a formal semantics. The translation of default theories into temporal logic provides default logic with a (temporal) semantics (both linear and branching time).

Furthermore, it is possible to describe 'mixed' forms of reasoning. For example, an agent may be described which uses default rules and auto-epistemic rules in the same line of reasoning.

We have given temporal interpretations of a number of well-known nonmonotonic reasoning methods, such as default logic, autoepistemic logic and logic programming. These interpretations were preserving in the sense that the limit models of the temporal interpretation corresponded to the classical static interpretations. Thus, in a sense we have embedded these logics in MTEL (MTEL*). There are other logics into which we can embed these nonmonotonic approaches. Default logic, autoepistemic logic and some semantics of logic programming can also be embedded into GK. In fact, we can use complexity results to show that any logic complete for the complexity class in which many nonmonotonic formalisms reside (see Chapter 9.1) has the property that all these formalisms can be embedded (polynomially) into it. The added value of our temporal logics is that they make *explicit* the (semi-)constructive nature of reasoning, often *implicitly* present.

This interest in the dynamics of reasoning is also present in two other approaches. One of them is the interpretation of logic programs in the situation calculus of Lin and Reiter ([LR96]). The basic idea is that the application of a program clause is an action in the situation calculus, with the effect of enlarging the set of known literals. The frame problem (facts only become known as an effect of a program clause application action, otherwise they remain as they were) is then solved using the technique of [Rei91]. The interesting thing is to see how a program clause is translated into the situation calculus. First of all, every predicate $F(\vec{x})$ is translated to a situation calculus predicate $F(\vec{x}, s)$ so that a predicate can be true or not *in a situation.* A situation in the situation calculus should be compared to a point in time in our own temporal interpretation of logic programs. A clause of the form

$$F(\vec{x}) \ :- G$$

where G is a sequence of predicates and negated (weakly) predicates which may contain variables, is translated into

$$G[s] \to F(\vec{x}, do(A(\vec{x}), s)).$$

Here $G[s]$ is the translation of G (we will come back to this), A is the *name* of the action of applying this sequent (every action must have a name in the situation calculus), and *do* is the function that returns the new situation that results from the application of an action in a situation. If we interpret $do(A(\vec{x}), s)$ as the *next* situation, then this translation is very similar to our temporal form

$$\text{conditions} \to X \text{ conclusions.}$$

In fact, this similarity goes even further if we consider the translation of G. Not bothering with the variables (we can always take ground instantiations), suppose $G = a_1, \ldots, a_n, -b_1, \ldots, -b_m$, where $-$ stands for weak negation (usually written `not`). Then the translation $G[s]$ is

$$G[s] = a_1(s) \wedge \ldots \wedge a_n(s) \wedge \neg(\exists s')b_1(s') \wedge \ldots \wedge \neg(\exists s')b_m(s').$$

The parts $\neg(\exists s')b_1(s')$ play the role of looking into the future (some axioms are used for the situation calculus which ensure that all situations are the effect of applying actions to the initial state) and are analogous to our $\neg FK(b_1)$! One difference between their approach and ours is that they use situations, and we use natural numbers. In fact, this probably makes our approach more similar to the approach of Wallace ([Wal93]), also discussed in [LR96]. One proof of the usefulness of their approach, is that dynamic control aspects can be modeled (which is also one of the aspects of reasoning we hope to be able to model using temporal logic). This is shown, for example, by the ability to model the Prolog cut operator (`!`) in the situation calculus ([Lin97]). In this formalization, the ability to name the action of applying a clause, is used in an essential manner. Since our temporal interpretation does not use explicit names for the action of applying a sequent, it would be interesting to see if we can still model the cut operator in our interpretation.

The second approach based on ideas similar to our own, is step-logic (now called *active logic* in general; see [Elg88, EP90, NKP94]). To quote from [NKP94]:

> Most common sense reasoning formalisms do not account for the passage of time as the reasoning occurs, and hence are inadequate from the point of view of modeling an agent's *ongoing* process of reasoning.

One of their interests is in modeling *limited reasoning*, in which an agent does not necessarily (immediately) know all logical consequences of its knowledge; but reasoning about the time left to do more reasoning is another possibility. Their model of time is the set of natural numbers, and the most important objects in step-logic are a kind of time sensitive inference rules. A good example of such a rule is the interpretation of modus ponens:

$$\frac{i : \ldots, \alpha, \alpha \to \beta}{i+1 : \ldots, \beta}$$

The interpretation is that for any time point i, if α and $\alpha \to \beta$ have been derived at that time point, then β is derived at time point $i + 1$. Note the similarity with our temporal interpretation of modus ponens in Section 5.4:

$$C(\mathsf{at}_\alpha) \wedge C(\mathsf{at}_{\alpha\to\beta}) \to \exists \mathrm{X} C(\mathsf{at}_\beta).$$

Step-logic is essentially a syntactical framework (but see [NKP94]), and the set of formulae derived at some point in time is in general not closed under provability; in our semantical approach, we have to make a change in signature, introducing atoms of the form at_α, in order to achieve this. Another example is a rule allowing negative introspection (which takes one step in time). One can also define the following interesting rule:

$$\frac{i : \ldots}{i+1 : \ldots , Now(i+1)}$$

which allows the agent to look at the clock. Another interesting rule is inheritance, which corresponds to conservativity in our framework:

$$\frac{i : \ldots , \alpha}{i+1 : \ldots , \alpha}$$

which should of course never hold for the *Now* predicate. One last sort of rule is the observation rule:

$$\frac{i : \ldots}{i+1 : \ldots , \alpha}$$

if α is observed at time $i+1$. The formulae that are observed at time points are represented by a function *OBS* mapping time points to finite sets of formulae. In our framework, we would represent this by rules of the form

$$at_i \rightarrow \exists \mathrm{X}(K\alpha)$$

where at_i is a formula true exactly at time point i in all temporal models (see Definition 9.19). Alternatively, we could use $\forall\mathrm{X}$ if we want the observation to be made always. Formally, an inference function *INF* is defined which takes a history (a finite sequence of states, where a state is a pair of finite sets, one holding the observations at that time point, one holding the derived formulae of that time point) to a finite set of (new) histories which are one-step extensions of the original. This means that the construction of histories is a truly constructive process: the rules may refer to the history, but never to the future. This means that a temporal translation of Reiter's default logic as we have done in Sections 5.1 and 5.2 is not possible in step-logic. The way nonmonotonic reasoning of the agent can be modeled is by modeling the backtracking behavior of the agent in search of the 'correct' default extensions (this in fact models default reasoning at a lower level of abstraction than we have done in this chapter). In step-logic, the agent can apply a default, and later find out that that leads to inconsistency (which can be handled in the syntactic states), and then 'forget' the conclusion of the default and try other possibilities. Step-logic is more general in that it is first-order and has some features (like looking at the clock or making observations) that we did not treat in Section 5.4. These latter features could also be defined. In fact, the temporal inference rules of step-logic can in general be translated into formulae of temporal partial logic (certainly the ones only referring to i and $i+1$, but one can think also of more difficult rules using complex temporal expressions). And in the temporal domain, our approach is more general, since we can also refer to the future (their temporal format is a restriction of ours).

Acknowledgments

The temporal interpretation of default logic in linear time epistemic logic appeared in [ET93]. The branching time approach was given in [ET96c]; preliminary results were described in [ET94b], where also the temporal interpretation of proof systems was given. The results on logic programs with stable generated models appeared in [EH97], along with some of the results on disjunctive rules.

Chapter 6

Execution of Temporal Theories

In the previous chapter, it was shown that many forms of reasoning can be specified in some temporal logic of information. An execution mechanism for such specifications would thus yield a general implementation for reasoning patterns. So what does execution of a temporal theory mean? Execution of a theory should lead to behavior which satisfies the theory. Thus, execution should yield a model of the theory. It is sometimes required that the execution mechanism can, in principle, find all models of the theory. A temporal logic with such an execution algorithm is called an *executable temporal logic* (see [BFG$^+$96]). In this section, executability of a class of temporal theories is discussed. First, an algorithm is given which can find minimal models of temporal theories of a restricted form. Then, a compositional reasoning system is described which implements this algorithm.

6.1 An algorithm for executing theories of reasoning

The algorithm to be presented below will find minimal models of a temporal theory. As the temporal language, we will take TEL, and as model class the class of closed conservative TEL models. The preference ordering we will take is a slight variation on $\preceq^g$. The ordering $\preceq^g$ prefers models in which the overall knowledge is minimal, the first point in time included. This means that also the knowledge at time point 0 is minimized. We introduce an ordering here which only compares models with the same initial state. That is, for a fixed initial state, the subsequent knowledge over time is minimized. (There is no inherent reason to prefer either of the two orderings. We have chosen this variant here to retain compatibility with [ET96a].)

Definition 6.1 The ordering $\preceq_0^g$ on closed TELC models is defined by

$$\mathcal{M} \preceq_0^g \mathcal{N} \Leftrightarrow \mathcal{M}_0 = \mathcal{N}_0 \text{ and } \mathcal{M} \preceq^g \mathcal{N}.$$

To give an example of how this ordering works, if we take the translation τ of default rules into TEL-formulae defined earlier, then the $\preceq_0^g$-minimal models of $\tau(D)$, for a set of default rules D, correspond to extensions of default theories $\langle D, W\rangle$, for all possible W. Or, in this preferential temporal logic, a set of defaults D specifies the reasoning frame operator Tr_D of Section 3.1 in accordance with Definition 4.63.

We will restrict ourselves to a specific kind of theories, consisting of rules very similar to the reasoning formulae of Section 4.5. The correctness of the algorithm is in fact based on a characterization result very similar to those presented in that section.

Apart from the temporal operators already introduced, we define an additional operator that refers to the first time point.

Definition 6.2 The temporal operator H_0 is defined by

$$(\mathcal{M}, t) \models H_0\varphi \Leftrightarrow (\mathcal{M}, 0) \models \varphi.$$

The new operator was in fact already expressible, since $H_0\varphi$ is equivalent to $\Box(H(\bot) \to \varphi)$. We are now ready to introduce theories of reasoning.

Definition 6.3 (Theory of Reasoning) A *theory of reasoning* is a set consisting of temporal formulae of the form $\alpha \wedge \beta \wedge \varphi \wedge \psi \to X(K\gamma)$, where

$\alpha = \bigwedge\{H_0(K\epsilon) \mid \epsilon \in A\}$ for a finite set of propositional formulae A,
$\beta = \bigwedge\{\neg H_0(K\delta) \mid \delta \in B\}$ for a finite set of propositional formulae B,
$\varphi = \bigwedge\{\neg F(K\theta) \mid \theta \in D\}$ for a finite set of propositional formulae D,
$\psi = \bigwedge\{K\zeta \mid \zeta \in F\}$ for a finite set of propositional formulae F, and
γ is a propositional formula.

These theories of reasoning will be studied further in Section 7.1.

In order to execute a theory of reasoning we interpret its temporal rules as inference rules. If the condition of such a rule is met, we introduce its conclusion at the next step. The condition of the rule pertaining to the initial facts and the present can be checked in a straightforward manner. The only problem is the part referring to the future, φ. The way to deal with them is to either a) assume they will be met and add the conclusion; in this case we will have to check at all later steps that they are indeed met, or b) assume they are not met; in this case we do not add the conclusion. In this case φ must be violated at some later time. If this does not happen than the execution is not correct. If the theory of reasoning is infinite, it is in general not possible at any point in time to be sure we are executing correctly.

Notice that a reasoning formula is not of the form "past implies future", a form used often for executable temporal logic (see [Gab89]). Of course we could move the future part to the right of the implication. Using the notation of Definition 6.3 we would obtain a rule

$$\alpha \wedge \beta \wedge \psi \rightarrow X(K\gamma) \vee \bigvee\{F(K\theta) \mid \theta \in D\}$$

which is clearly in the desired format. To execute this rule if the conditions are met, we could either introduce γ at the next moment in time, or introduce one of the elements of D at any future time point. But this is not the correct way of executing such a formula, since the conjuncts of φ are meant as a kind of consistency check: if $\neg F(K\theta)$ is true, then $\neg\theta$ remains consistent with the agent's beliefs. So they should be *declarative* and not *imperative*. So instead of the slogan "declarative past implies imperative future" ([Gab89]) we use the slogan "declarative past and future imply imperative future".

We will now informally describe the general algorithm for executing a theory of reasoning Th. We use predicates $never\_true(\alpha)$ and $next(\alpha)$ where α is propositional, and $sometimes\_true(D)$, where D is a finite set of propositional formulae. Intuitively, the predicate $never\_true(\alpha)$ expresses a constraint over the future that α may not become true, the predicate $sometimes\_true(D)$ expresses a constraint over the future that at least one of the formulae in D should become true in the future, and the predicate $next(\alpha)$ expresses the fact that we should add the formula α to the set of conclusions at the next point in time.

We assume that we have a set of initial facts.

Algorithm 6.4

1. Mark all rules as unused, set all predicates to false, and set t to 0.
2. If the current facts are contradictory, backtrack to the previous time point.
3. Check all constraints $never\_true(\theta)$. If θ is entailed by the current facts, backtrack to the previous time point.
4. For each unused rule

   $$\alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma),$$

   where

   $\alpha = \bigwedge\{H_0(K\epsilon) \mid \epsilon \in A\}$ for a finite set of propositional formulae A,
   $\beta = \bigwedge\{\neg H_0(K\delta) \mid \delta \in B\}$ for a finite set of propositional formulae B,
   $\varphi = \bigwedge\{\neg F(K\theta) \mid \theta \in D\}$ for a finite set of propositional formulae D,
   $\psi = \bigwedge\{K\zeta \mid \zeta \in F\}$ for a finite set of propositional formulae F, and
   γ is a propositional formula;

   do:
   If all formulae in F are entailed by the current facts, and all formulae in A are entailed by the initial facts, and all formulae in B are not entailed by the initial facts then this rule is applicable: mark this rule as *used* and do either of:

- introduce $next(\gamma)$ and constraints $never\_true(\theta)$ for each $\theta \in D$, or
- introduce a constraint $sometimes\_true(D)$.

If we backtracked to this time point, make a choice for all the applicable rules at this time point which has not been made before. If this is not possible, backtrack one more step. If this is not possible ($t = 0$) then abort.

5. $t := t + 1$; If there are formulae of the form $next(\gamma)$, then introduce γ and delete $next(\gamma)$ for all of them; goto 2. If there are no such formulae, then check for each constraint $sometimes\_true(D)$ whether some $\theta \in D$ is entailed by the current facts; if not, backtrack one step. Otherwise a minimal model has been found.

   If for all constraints $sometimes\_true(D)$, some $\theta \in D$ is entailed by the facts at some point in time, the execution of the algorithm is called *correct*.

If at each step new formulae are added, we can never be sure during execution, whether the execution will be correct. However, in case the theory of reasoning is finite, there will be a point in time when no new facts are added: then the execution will be correct. This is always the case when the propositional language contains a finite number of atoms: then there is a finite number of non-equivalent propositional formulae; if this number is n, then after n steps of the algorithm, no new facts can be added. For correct executions we have the following:

Theorem 6.5 Let a theory of reasoning Th be given. For an execution of the algorithm for Th, let T_i denote the set of propositional formulae derived at time point i. Define a reasoning trace $(\mathcal{M}_i)_{i\in\mathbb{N}}$ by $\mathcal{M}_i = Mod(T_i)$. Let $\mathcal{T}$ be the set of these traces for all possible correct executions of the algorithm for Th. Then $\mathcal{T}$ is exactly the set of $\preceq_0^g$-minimal models of Th.

We will not give a detailed proof of this theorem. There is a characterization theorem for $\preceq_0^g$-minimal models very similar to Corollary 4.55 (the condition on the first state has to be dropped). The algorithm finds the models thus characterized by a kind of depth-first search process.

The generality of this algorithm depends on the generality of reasoning theories as a specification language, studied in the next chapter (Section 7.1). Applying this algorithm to the theory of reasoning of a default theory (where the initial facts are the axioms), we obtain an algorithm very similar to the ones meant especially for default logic: it picks a subset of so-called *generating* defaults, and checks whether it indeed induces an extension.

A special class of theories of reasoning is obtained when we do not allow consistency checks (the set D in Definition 6.3 is empty). The induced reasoning process will then be monotonic, and the algorithm will be much more efficient.

In specific instances of theories of reasoning one would like to make the algorithm more efficient. This can be done by using heuristic knowledge to make smart choices at each point in time. In particular, if at the current point in time one of the $\theta \in D$ is already entailed by the facts, only the second choice in the algorithm makes sense. In the case of default logic, one could use priorities between default rules or specificity of rules to restrict the number of possible choices (see [Bre94a]). The set of possible runs of our algorithm can be parameterized by selection functions. Such functions describe the choices which have to be made at each point in time (in a similar fashion as in [TT92]). Then the "good" selection functions make use of heuristic knowledge to guide the reasoning process.

6.2 A compositional reasoning system for executing theories of reasoning

Implementations of agents that can reason in a defeasible manner need to include an implemented nonmonotonic reasoning system. In different applications, agents may need different types of nonmonotonic reasoning. Therefore, it is useful to develop a generic reasoning system that covers different types of nonmonotonic reasoning (as opposed to implementations for one specific nonmonotonic formalism as in e.g., [Nie96], [RS94] or [CMT96]). This can be done by a reasoning system that executes theories of reasoning, which may describe different forms of reasoning. The development of such a generic reasoning system can be made in a transparent manner if a central role is played by an implementation-independent design specification based on current software engineering principles, such as compositionality and information hiding. In this section, a compositional reasoning system is introduced which implements Algorithm 6.4.

The system was developed using the compositional modeling environment DESIRE (framework for DEsign and Specification of Interacting REasoning components; see [BJT98, BDJT95, BTWW95]). This environment for the development of compositional reasoning systems and multi-agent systems has been successfully used to design various types of reasoning systems and multi-agent systems and applications in particular domains. The DESIRE software environment offers a graphical editor, an implementation generator and an execution environment to execute specifications automatically. In its use to build complex reasoning systems, DESIRE can be viewed as an advanced theorem proving environment in which both the knowledge and the control of the reasoning can be specified in an explicit, declarative and compositional manner. We will give a brief description of this framework.

6.2.1 A specification framework for compositional systems

The basic structure of a (compositional) multi-agent system in DESIRE is a hierarchical structure of *components*, in which each component is assigned a *task* or *process* to perform. To give an example, we may have a top component diagnosis, which is a process performing diagnosis. The task 'diagnosis' could be composed of two subtasks: determining a candidate hypothesis (a disease, or a malfunctioning part of a system), and then testing that hypothesis. Our component diagnosis may have subcomponents hypothesis_determination and hypothesis_validation which perform these respective tasks. The latter component may again contain some subcomponents performing the (possible) subtasks of selecting observations to make, performing these observations, and interpreting the results. Whether a component counts as an agent depends on a number of characteristics: its autonomy, social ability, reactivity, and pro-activeness ([WJ95]). The top component, diagnosis, in this example, might be an agent able to communicate with other agents, which completely controls the execution of its subcomponents, which should not be considered to be agents.

Components of course do not work in isolation, they have to be able to exchange information. To this end, all components have an input interface and an output interface, which contains facts in some input, respectively output, language. These facts are essentially atomic statements in a predicate language, along with their truth value: true, false or unknown. The information exchange takes place by activating *information links*, which transfer facts from one component's interface to another component's interface (possibly performing a translation: different interfaces may use different languages). Given a component C, an information link of C may transfer information from C's input interface to the input interface of a subcomponent, or from the output interface of a subcomponent to C's output interface, or, lastly, from the output interface of a subcomponent to the input interface of (usually another) subcomponent. To control the activation of these links, C contains *task control knowledge*, which also contains knowledge about when and how to activate C's subcomponents. Of course, components at the lowest level do not contain subcomponents. Such components are called *primitive* (in contrast to the others, which are called *composed*), and usually contain a knowledge base.

In order to structure the process of building such a system, in the framework DESIRE knowledge of process composition and knowledge composition is explicitly modeled and specified. To give the reader a structured but concise view of these types of knowledge, we quote a relevant portion from [BJT98]. The quote (which is not in quotation marks) ranges from the next paragraph until the last paragraph before Subsection 6.2.2.

6.2.1.1 Process composition

Process composition identifies the relevant processes at different levels of (process) abstraction, and describes how a process can be defined in terms of lower level processes.

6.2.1.1.1 Identification of processes at different levels of abstraction Processes can be described at different levels of abstraction; for example, the processes for the multi-agent system as a whole, processes within individual agents and the external world, processes within task-related components of individual agents. Different views can be taken: a task perspective, and a multi-agent perspective. The *task perspective* refers to the view in which the processes needed to perform an overall task are distinguished. These processes (or sub-tasks) are then *delegated* to appropriate agents and the external world. The *multi-agent perspective* refers to the view in which agents and one or more external worlds are first distinguished and then the processes within them, including agent-related processes such as management of communication, or controlling its own processes.

Specification of a process The identified processes are modeled as *components*. For each process the *types of information* required as input, and resulting as output, are identified as well. This is modeled as *input and output interfaces* of the components.

Specification of abstraction levels The identified levels of process abstraction are modeled as *abstraction / specialization relations* between components at adjacent levels of abstraction: components may be *composed* of other components or they may be *primitive*. Primitive components may be either reasoning components (for example based on a knowledge base), or, alternatively, components capable of performing tasks such as calculation, information retrieval, optimization, et cetera. The identification of processes at different abstraction levels results in specification of components that can be used as building blocks, and of a specification of the sub-component relation, defining which components are a sub-component of a which other component. The distinction of different process abstraction levels results in process hiding.

6.2.1.1.2 Composition relation for processes The way in which processes at one level of abstraction are composed of processes at the adjacent lower abstraction level is called *composition*. This composition of processes is described by the possibilities for *information exchange* between processes (*static view* on the composition), and *task control knowledge* used to control processes and information exchange (*dynamic view* on the composition).

Information exchange Knowledge of information exchange defines which types of information can be transferred between components and the *information links* by which this can be achieved.

Task control knowledge Components may be activated sequentially or they may be continually capable of processing new input as soon as it arrives (awake). The same holds for information links: information links may be explicitly activated or they may be awake. Task control knowledge specifies under which conditions which components and information links are active (or awake). Evaluation criteria, expressed in terms of the evaluation of the results (success or failure), provide a means to guide further processing. Task control knowledge can be specified to constrain the number of possible process traces that can be generated. Depending on the application, task control knowledge can be specified in different ways, varying from rather open approaches that entail almost no constraints on the behavior (e.g., when all components and links are made awake), to a strictly prescribed sequence of activations of components and links. Task control is specified separately for each process abstraction level. The degree to which behavior is constrained by task control can differ for these abstraction levels, and can differ between components within one abstraction level. For example, at the top level of a system, agents and the links between agents may be not constrained in their behavior (which realizes their autonomy), and within an agent at a lower process abstraction level, task control may specify a fixed sequence of activation of components and links.

6.2.1.2 Knowledge composition

Knowledge composition identifies the knowledge structures at different levels of (knowledge) abstraction, and describes how a knowledge structure can be defined in terms of lower level knowledge structures. The knowledge abstraction levels may correspond to the process abstraction levels, but this is not often the case; often the matrix depicted in Figure 6.1 shows more than a one to one correspondence between process abstraction levels and knowledge abstraction levels.

6.2.1.2.1 Identification of knowledge structures at different abstraction levels The two main structures used as building blocks to model knowledge are: *information types* and *knowledge bases*. Knowledge structures can be identified and described at different levels of abstraction. At the higher levels the details can be hidden. The resulting levels of knowledge abstraction can be distinguished for both information types and knowledge bases.

Information types An information type defines an ontology (lexicon, vocabulary) to describe objects or terms, their sorts, and the relations or functions that can be defined on these objects.

Knowledge bases A knowledge base defines a part of the knowledge that is used in one or more of the processes. Knowledge bases use ontologies defined in information types. Which information types are used in a knowledge base defines a relation between information types and knowledge bases.

6.2.1.2.2 Composition relation for knowledge structures Information types can be composed of more specific information types, following the principle of compositionality discussed above. Similarly, knowledge bases can be composed of more specific knowledge bases. The compositional structure is based on the different levels of knowledge abstraction that are distinguished, and results in information and knowledge hiding.

6.2.1.3 Relation between process composition and knowledge composition

As shown in Figure 6.1, *compositionality of processes* and *compositionality of knowledge* are two different dimensions. Each process in a process composition uses knowledge structures. Which knowledge structures are used for which processes is defined by the relation between process composition and knowledge composition. The compositional knowledge structures can be related to one or more compositional process structures, where needed; the cells within the matrix depicted in Figure 6.1 define these relations. Note that not all cells need to be filled in this matrix. For example, in a special case where knowledge composition is completely dependent on the process composition the matrix in Figure 6.1 shows only a diagonal of filled cells.

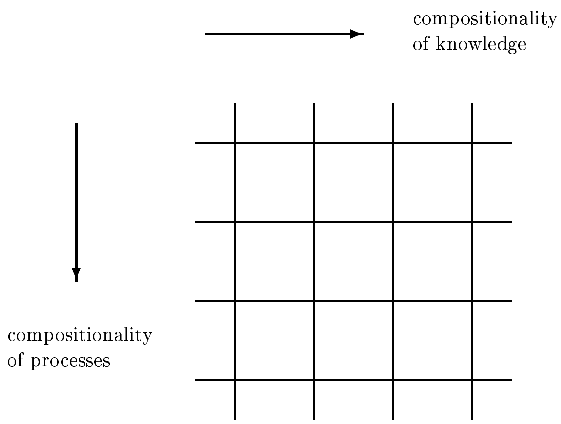

Figure 6.1: Compositionality of processes and compositionality of knowledge.

For a more thorough and detailed description of compositional design using DESIRE, we refer the interested reader to [BDJT95, BJT98, BTWW96].

6.2.2 A generic compositional nonmonotonic reasoning system

The system described here will try to find a minimal model of a given theory. The system at present only handles rules in which all propositional formulae are actually literals, so no 'real' propositional reasoning is necessary. The system searches for a model by incrementally trying to find a next state for a finite initial part of a (possible) model. The reasoning task of finding minimal models of a reasoning theory is modeled as a composition of two subtasks. The first subtask generates all possible continuations (of one state) of the current initial part of a model, and the second subtask selects a continuation. Within the first task, first it is determined which rules are applicable. Applicable rules are the rules

$$\alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma),$$

where

$\alpha = \bigwedge\{H_0(K\epsilon) \mid \epsilon \in A\}$ for a set of propositional formulae A,
$\beta = \bigwedge\{\neg H_0(K\delta) \mid \delta \in B\}$ for a set of propositional formulae B,
$\varphi = \bigwedge\{\neg F(K\theta) \mid \theta \in C\}$ for a set of propositional formulae C,
$\psi = \bigwedge\{K\zeta \mid \zeta \in D\}$ for a set of propositional formulae D, and
γ is a propositional formula.

for which the conditions that refer to the past and present (i.e., α, β , ψ) are fulfilled.

Next, for each of the applicable rules, for the future-directed conditions φ two possibilities are generated:

- either the conditions that refer to the future will be fulfilled in the reasoning trace that is generated, or
- these future-directed conditions will not be fulfilled.

In the first case the rule will contribute its conclusion to the reasoning process and we have to make sure in the future that the future conditions were indeed fulfilled (we add constraints to ensure this). In the second case no explicit contribution will be made by the rule. However, in this second case, by the subsequent generation of the reasoning trace it will have to be guaranteed that the future-directed conditions indeed will be violated (and we again add constraints to ensure this).

The design of the compositional nonmonotonic reasoning system has been specified in DESIRE, according to the five types of knowledge discussed earlier. Five levels of abstraction are distinguished in the task hierarchy (see Figure 6.2).

6.2.2.1 Top level of the system

At the highest level the system consists of four components. During the reasoning process, in the component maintain_current_state the facts are represented that have been derived. The component maintain_history stores relevant aspects of the reasoning process in order to perform belief revision if required. The reasoning is performed by the component generate_possible_continuations, which generates the possible next steps of the reasoning trace and select_continuation which chooses one of these possibilities. By this selection the actual next step in the reasoning trace is determined. In Figure 6.3 the information exchange at the top level of the system is depicted. In this picture and the following ones, a rectangle is a component, and the arrows are information links. The small rectangles on the left and right of the components are the input, respectively output, interfaces. (In these pictures we have left out some information links which go from a component to itself. For instance, such links are sometimes needed to model a closed-world assumption.)

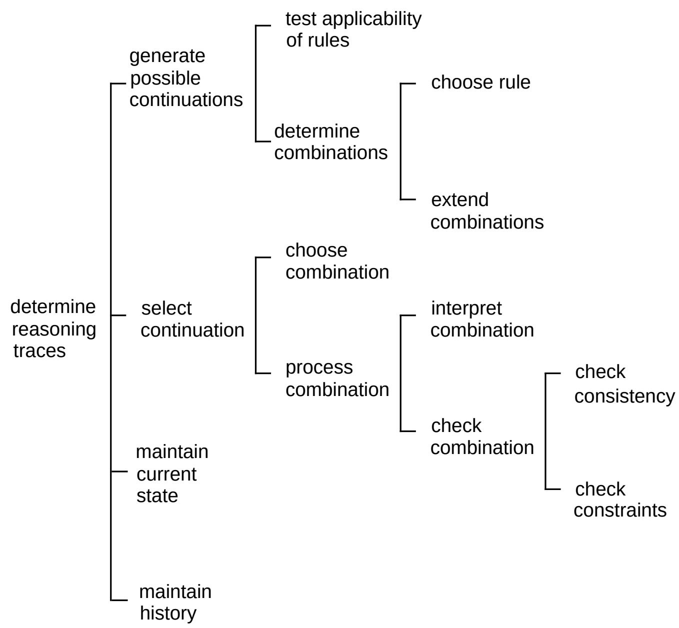

Figure 6.2: Complete task hierarchy of the system.

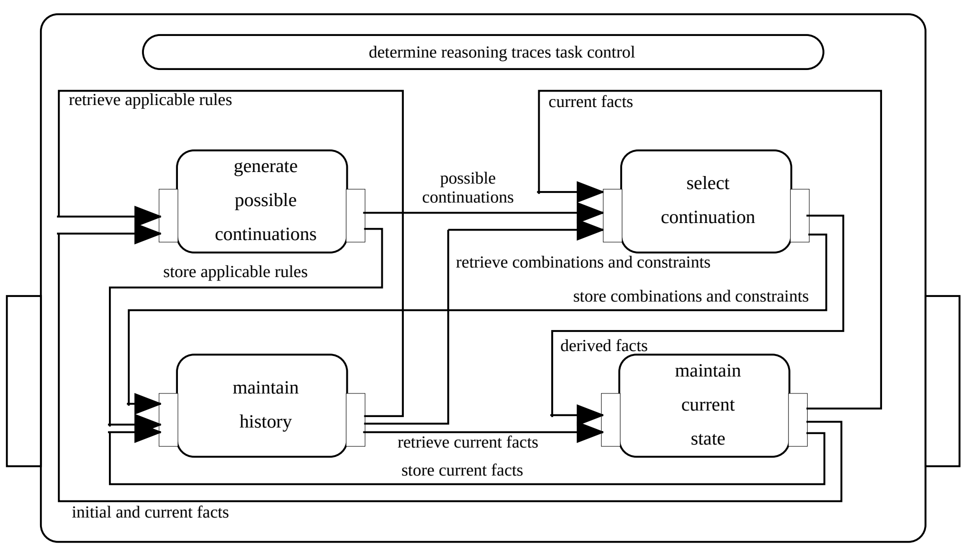

Figure 6.3: Information exchange at the top level of the system.

6.2.2.2 Generate possible continuations

Within the component generate_possible_continuations two subcomponents are distinguished: test_applicability_of_rules and determine_combinations; see Figure 6.4 for the information flow at this level.

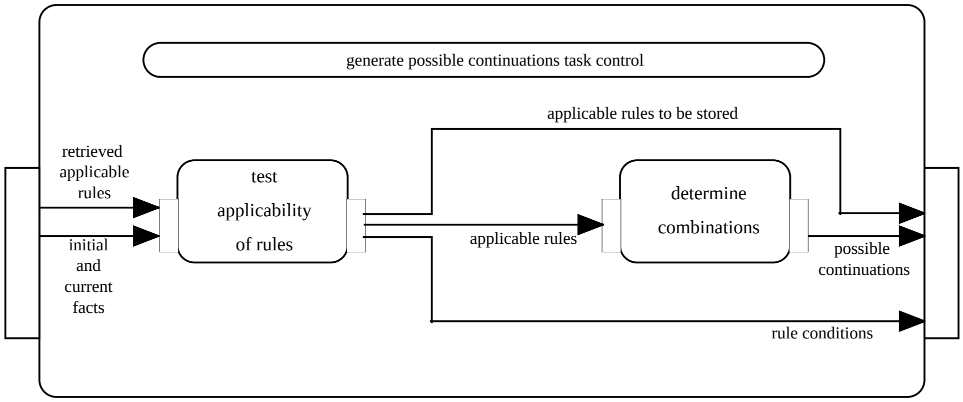

Figure 6.4: Information exchange within generate_possible_continuations.

The former component is primitive and determines the rules that are applicable in the current state of the reasoning process. Its knowledge base consists of rules, for example of the form:

if pos_H0_condition(L: Literals, R: Rules)	**if** current_condition(L: Literals, R: Rules)
and not initial_fact(L: Literals)	**and not** current_fact(L: Literals)
then not applicable(R: Rules)	**then not** applicable(R: Rules)

By application of a form of the closed world assumption, the applicable rules are derived. The second component determines for each of the applicable rules two possibilities: it is assumed that either (+) the conditions of the rule that refer to the future will be fulfilled in the reasoning trace that is generated, or (-) these future-directed conditions will not be fulfilled.

The component determine_combinations is rather simple (see Figure 6.5). The applicable rules are treated one by one and combinations are constructed. Both subcomponents are primitive. The knowledge base of the first subcomponent, choose_rule just consists of one rule:

if applicable(R: Rules)
and not covered(R: Rules)
then in_focus(R: Rules)

This means that any applicable rule that has not been chosen before, can be deduced to be chosen (in_focus). By invoking this component in such a way that it will only deduce one new fact, one new rule is chosen at every invocation.

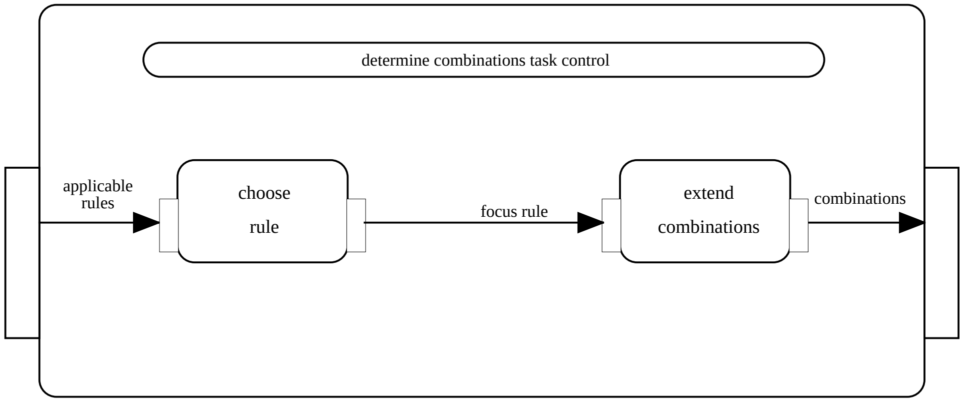

Figure 6.5: Information exchange within determine_combinations.

The knowledge base of the second subcomponent, extend_combinations consists of the following two rules:

if old_combination(C: combinations)
and in_focus(R: rules)
and future_dependent(R: rules)
then new_combination(
app(C: combinations, tup(R: rules, neg)));

if old_combination(C: combinations)
and in_focus(R: rules)
then new_combination(
app(C: combinations, tup(R: rules, pos)));

These rules build new combinations from old combinations and the rule in focus. The predicate app (for *append*) is used to build up a list, and the predicate tup (for *tuple*) is used to make tuples. Only rules with a part that refers to the future (future_dependent) are allowed *not* to be applied (meaning that tup(R: rules, neg) may occur in a combination). Which rules are future_dependent is deduced in test_applicability_of_rules.

6.2.2.3 Select continuation

Within the component select_continuation, focus combinations are chosen one by one and processed (see Figure 6.6).

Processing a combination is performed by first making an interpretation of the information represented by a combination, and subsequently checking on consistency against the current facts and checking the constraints (see Figures 6.7 and 6.8).

The knowledge base of the primitive component check_consistency consists of three rules, an example of which is:

if next(A: Atoms)
and current(neg(A: Atoms))
then inconsistency

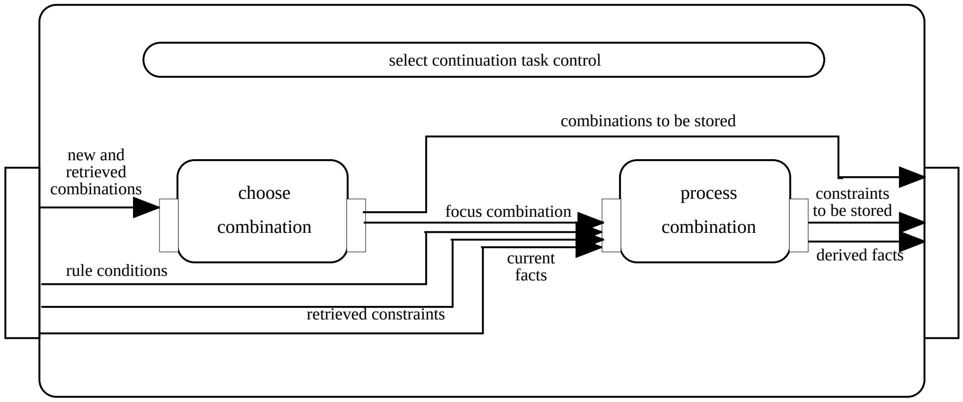

Figure 6.6: Information exchange within select_continuation.

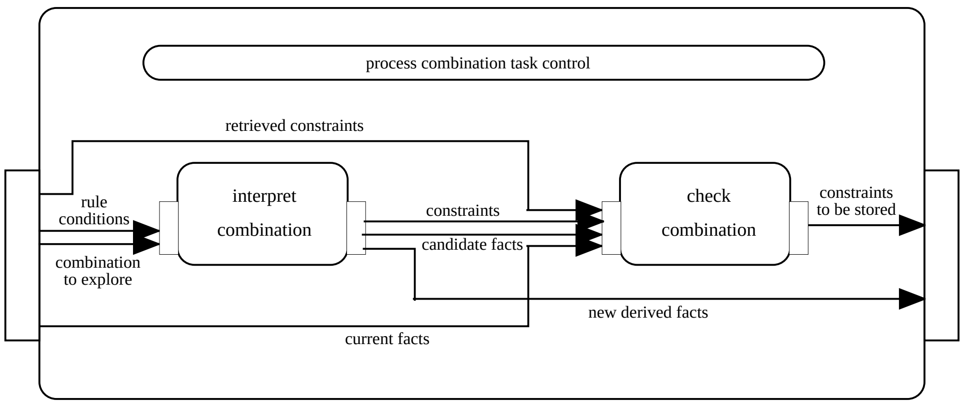

Figure 6.7: Information exchange within process_combinations.

6.2.3 Example trace

In this subsection we will give an example of the execution of our generic reasoning system. The theory of reasoning is the translation of the following example default theory $\langle D, W\rangle$ with $D = \{(a : b)/b, (d : c)/c, (b : \neg c)/e\}$ and $W = \{a, d, b \rightarrow \neg c\}$. (Notice that this is the example default theory of Example 5.2.) The atoms a and d will be initial facts. Formally, the formula $b \rightarrow \neg c$ should also be a fact, but since the generic reasoning system to be described below does not perform general propositional reasoning (it uses a subset of natural deduction called *chaining*), we will translate it into the following two rules which describe the application of the formula:

1. $Kb \rightarrow X(K\neg c)$,

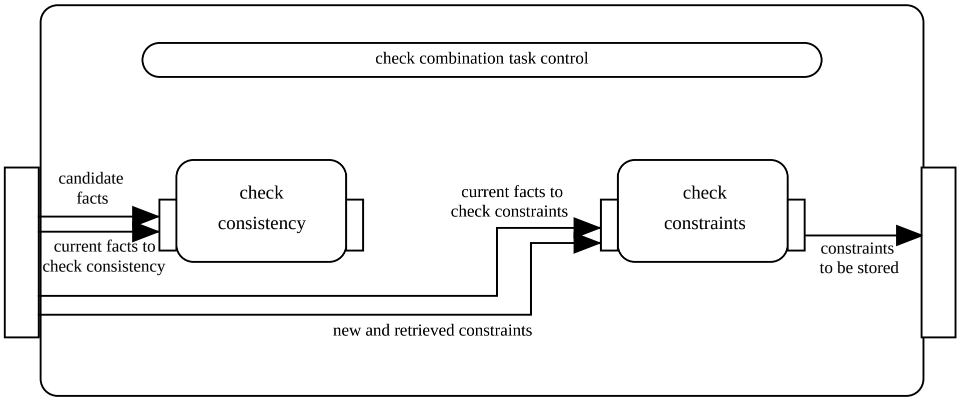

Figure 6.8: Information exchange within check_combinations.

2. $Kc \rightarrow X(K\neg b)$.

The default rules are translated as follows:

3. $Ka \wedge \neg F(K\neg b) \rightarrow X(Kb)$,

4. $Kd \wedge \neg F(K\neg c) \rightarrow X(Kc)$,

5. $Kb \wedge \neg F(Kc) \rightarrow X(Ke)$.

When this theory is given to the generic reasoning system, the following will happen (in this list, on the left we will indicate the facts that are derived, on the right the component in which it is derived, in brackets; not everything derived in every component is listed and no information links are listed):

-	a, d	(maintain_current_state)
-	applicable(r3), applicable(r4)	(test_applicability_of_rules)
-	new_combination(app(app(nil, tup(r3, pos)), tup(r4, pos))), new_combination(app(app(nil, tup(r3, neg)), tup(r4, pos))), new_combination(app(app(nil, tup(r3, pos)), tup(r4, neg))), new_combination(app(app(nil, tup(r3, neg)), tup(r4, neg)))	(determine_combinations)
-	selected_combination(app(app(nil, tup(r3, pos)), tup(r4, pos)))	(choose_combination)
-	next(c), next(b)	(interpret_combination)
-	c, b	(maintain_current_state)
-	applicable(r1), applicable(r2)	(test_applicability_of_rules)
-	new_combination(app(app(nil, tup(r1, pos)), tup(r2, pos)))	(determine_combinations)

(since these rules do not refer to the future, only one combination will be generated, in which both rules are applied)

- selected_combination(app(app(nil, tup(r1, pos)), tup(r2, pos)))	(choose_combination)
- next(neg(b)), next(neg(c))	(interpret_combination)
- inconsistency	(check_consistency)

(this uses the rule **if** next(neg(A: atoms)) **and** current(A: atoms) **then** inconsistency with next(neg(b)) and current(b); now maintain_history will be invoked with target set retrieve. Then choose_combination will fail, since there was only one combination at this point — app(app(nil, tup(r1, pos)), tup(r2, pos)) — so another retrieve is performed.)

- selected_combination(app(app(nil, tup(r3, neg)), tup(r4, neg)))	(choose_combination)
- sometimes_true(neg(c), r4), sometimes_true(neg(b), r3)	(interpret_combination)

(after maintain_current_state, test_applicability_of_rules will find no more applicable rules).

- incorrect(r4)	(check_constraints)

(this is derived from the constraint sometimes_true(neg(c), r4)) and
not current(neg(c)); a retrieve will occur)

- selected_combination(app(app(nil, tup(r3, pos)), tup(r4, neg)))	(choose_combination)
- sometimes_true(neg(c), r4), next(b), never_true(neg(b))	(interpret_combination)
- b	(maintain_current_state)
- applicable(r1), applicable(r5)	(test_applicability_of_rules)
- new_combination(app(app(nil, tup(r5, pos)), tup(r1, pos))), new_combination(app(app(nil, tup(r5, neg)), tup(r1, pos)))	(determine_combinations)

(r1 has to be applied (it does not refer to the future), but for r5 there is a choice)

- selected_combination(app(app(nil, tup(r5, pos)), tup(r1, pos)))	(choose_combination)
- next(neg(c)), next(e), never_true(c)	(interpret_combination)
- neg(c), e	(maintain_current_state)

(no more applicable rules are found by test_applicability_of_rules, so check_constraints is invoked, which finds no violated constraints)
- at this point user_interaction, a component of the top level of the system left out so far, will display the window of Figure 6.9.

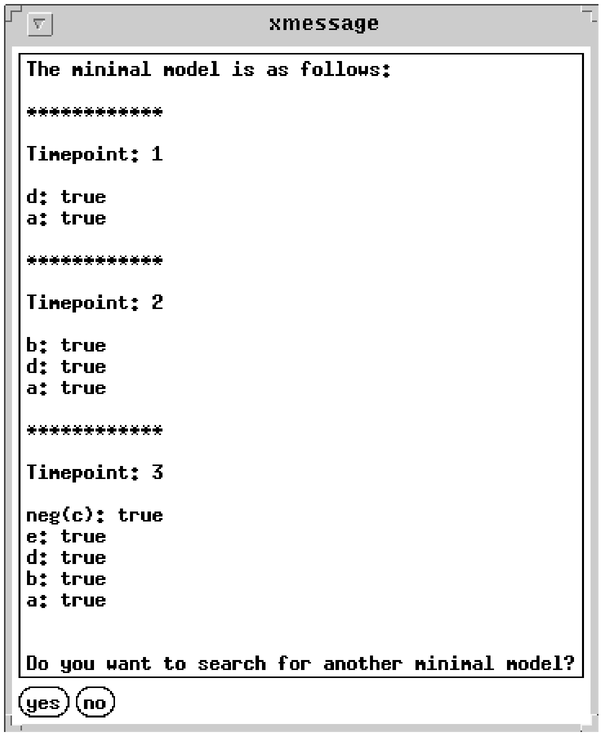

Figure 6.9: User interaction.

The minimal model displayed corresponds to the extension based on the literals $\{a, b, \neg c, d, e\}$. When the user clicks "no", the execution will stop. Otherwise a retrieve will be performed by maintain_history, after which choose_combination will choose the combination in which r5 is not applied (but r1 is). This will eventually lead to a violated constraint, so another retrieve will occur. Then choose_combination will take the last combination (for r3 and r4), in which r3 is not applied and r4 is. Ultimately, the second minimal model will be found and displayed (corresponding to the literals $\{a, \neg b, c, d\}$). If the user wants to search for another minimal model, none will be found, and a message indicating this will be displayed, after which execution ends.

6.3 Conclusions and related work

In this chapter we first described an algorithm to execute temporal theories of a certain format. Thus, our approach falls into the area of executable temporal logic (see for instance [Gab89, BFG$^+$96]), or more generally, executable modal logic (see

[FO95]). As mentioned before, the main difference with most other executable temporal logics is in the interpretation of the "$\neg FK$" part on the left hand side (or "FK" on the right hand side) of a rule, which we interpret declaratively, while the other approaches interpret it imperatively. Of course, our focus on modeling (possibly) non-monotonic reasoning is not shared by the other approaches. Furthermore, our states are three-valued information states, whereas the states are (two-valued) propositional valuations or assignments of values to parameters in most other approaches. One framework we would like to mention explicitly, is METATEM, which is the executable version of the temporal belief logic TBL already mentioned at the end of Chapter 4. The two main differences between METATEM and our approach (apart from the difference in states — three-valued in our approach, two-valued in METATEM) are the interpretation of the future operator (mentioned above), and the fact that an (information) state in METATEM holds the beliefs of more than one agent (whereas our logic is for one agent only, but see also Section 8.1). In the second section of this chapter, the framework DESIRE for the design of compositional reasoning systems and multi-agent systems was applied to build a generic nonmonotonic reasoning system. The main advantages of using DESIRE (compared to a direct implementation in a programming language such as PROLOG) are:

- the design is generic and has a transparent compositional structure; it is easily readable, modifiable and reusable. The generic nonmonotonic reasoning system is easily usable as a component in agents that are specified in DESIRE. For example, if an agent is designed that has default knowledge, then the generic reasoning system can be included as one of the agent's components and the representation of the agent's default knowledge can be translated to the temporal representation of the generic reasoning system.

- explicit declarative specification of both the static and dynamic aspects of the nonmonotonic reasoning processes is possible, including their control. The current system generates one or all reasoning traces that are possible without any specific guidance. However, a number of approaches to nonmonotonic reasoning have been developed that in addition use explicit knowledge about priorities between nonmonotonic rules (e.g., [Bre94a], [TT92]). This knowledge can easily be incorporated within the component select_continuation, in particular within its subcomponent choose_combination.

Even though the efficiency of the reasoning system can be improved (by adding knowledge to prevent generating possible continuations that can easily be seen to violate constraints, by adding heuristic knowledge in the selection of continuations, etc.), implementations for specific nonmonotonic formalisms (e.g. for default logic, see [Nie95], [CMT96], or [RS94]) can often be made more efficient. In general they lack, however, the ability to handle different kinds of nonmonotonic reasoning, and the extendibility of our approach. Also, they cannot handle dynamic queries (for example, the query whether a literal has been derived before time point 3).

Acknowledgments

The material in the first section appeared in [ET96a], and the material in the second section appeared in [ET97].

Chapter 7

Expressiveness

In Chapter 2, it was argued that reasoning can be formalized (by operators) on a number of levels of abstraction. These formalizations (operators) can be specified by specification languages. Chapter 3 introduced some of these languages, and a number of temporal specification languages were introduced in Chapter 4. When assessing a specification language, a major criterion is its expressiveness. There are at least two sides to expressiveness. On the one hand, one may ask whether it is easy to write down a specification in practice: does the language contain enough primitives and operators to allow you to conveniently write down what you intend? The temporal theories in Chapter 5 suggest that temporal logics of information have a good expressiveness for specifying reasoning behavior. On the other hand, there is a more theoretical concern to expressiveness: which objects can be specified in the language, and which objects cannot? The semantics of a specification language helps in answering the question: given a specification, which object does it denote? Here, we are interested in the reverse question: given an object, is there a specification whose meaning is that object?

This latter side to expressiveness is the concern of this chapter. Of course, all the different specification languages mentioned so far in this thesis, have their own expressiveness, and it would be a rather tiresome exercise to treat all of them. Therefore, in this chapter the expressiveness of two languages will be explored, one based on temporal logic, and one based on default logic.

7.1 Infinitary theories of reasoning

In this section, the expressiveness of infinitary theories of reasoning is investigated. An infinitary theory of reasoning is a variation on the theories of reasoning introduced in Chapter 6 using infinite conjunctions. The interpretation of an infinite conjunction in a (conservative closed) TEL-model is straightforward.

Definition 7.1 (Infinite conjunction) Let $\mathcal{M}$ be a closed conservative TEL-model, $t \in \mathbb{N}$, and let A be a set of TEL-formulae. Then the interpretation of the *infinite conjunction* of A, denoted $\bigwedge A$, is as follows:

$$(\mathcal{M}, t) \models \bigwedge A \Leftrightarrow (\mathcal{M}, t) \models \varphi \text{ for all } \varphi \in A.$$

The following definition of infinitary theories of reasoning is completely analogous to Definition 6.3, but without the word "finite".

Definition 7.2 (Infinitary theory of reasoning) An *infinitary theory of reasoning* is a set consisting of temporal formulae of the form $\alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma)$, where

$\alpha = \bigwedge\{H_0(K\epsilon) \mid \epsilon \in A\}$ for a set of propositional formulae A.
$\beta = \bigwedge\{\neg H_0(K\delta) \mid \delta \in B\}$ for a set of propositional formulae B.
$\varphi = \bigwedge\{\neg F(K\theta) \mid \theta \in D\}$ for a set of propositional formulae D.
$\psi = \bigwedge\{K\zeta \mid \zeta \in F\}$ for a set of propositional formulae F.
γ is a propositional formula.

If all conjuncts in all formulae are finite, the theory is called *finitary*.

The ordering on closed TELC models we use is the ordering $\preceq_0^g$ introduced in Chapter 6. Definition 4.63 tells us which reasoning frame operator is specified by an infinitary theory of reasoning S:

$$\mathcal{T}_S(X) = \{\mathcal{M} \mid \mathcal{M} \models_{\preceq_0^g} S \text{ and } Th(\mathcal{M}_0) = Cn(X)\}.$$

This reasoning frame operator is based on the information state frame IS^{ep} of epistemic states, with the restriction that the information states are *closed* sets of models. So the (reverse) expressiveness question now becomes: given a reasoning frame operator $\mathcal{R}$ based on this information state frame, does there exist an infinitary theory of reasoning S such that $\mathcal{T}_S = \mathcal{R}$?

The reasoning frame operators specified by an infinitary theory of reasoning are particularly well-behaved, in that they satisfy all the properties mentioned in Section 2.3. Moreover, these properties exactly characterize the reasoning frame operators that can be specified by an infinitary theory of reasoning, as the following theorem shows.

Theorem 7.3

1. For any infinitary theory of reasoning S, the associated reasoning frame operator $\mathcal{T}_S$ is conservative and eager, and satisfies non-inclusiveness, uniqueness of traces, and invariance (see Definition 2.21).

2. Suppose $\mathcal{R}$ is a conservative and eager reasoning frame operator satisfying non-inclusiveness, uniqueness of traces, and invariance. Then there exists an

infinitary theory of reasoning S such that $\mathcal{T}_S = \mathcal{R}$. If the language contains only finitely many propositional atoms, then S can be taken finite and finitary.

Proof: Throughout this proof, we will make use of the following characterization result for $\preceq_0^g$-minimal models of an infinitary theory of reasoning. This result is very similar to Corollary 4.55, and as the proof is also very similar, we leave it to the interested reader. The characterization is as follows: $\mathcal{M}$ is a $\preceq_0^g$-minimal model of an infinitary theory of reasoning S if and only if for each $s \in \mathbb{N}$, $\mathcal{M}_{s+1} = Mod(Th(\mathcal{M}_s) \cup \{\gamma \mid \text{there is a rule } \alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma) \in S \text{ such that } (\mathcal{M}, s) \models \alpha \wedge \beta \wedge \varphi \wedge \psi\})$. We will now proceed to prove the two claims.

1. Suppose we have an infinitary theory of reasoning S. Let us consider $\mathcal{T}_S$. We will prove the properties one by one.
   • *conservative:* Take any $\mathcal{M} \in \mathcal{T}_S(X)$, then $\mathcal{M}$ is a $\preceq_0^g$-minimal model of S, so for any $s \in \mathbb{N}$ we have $\mathcal{M}_{s+1} = Mod(Th(\mathcal{M}_s) \cup \{\gamma \mid \text{there is a rule } \alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma) \in S \text{ such that } (\mathcal{M}, s) \models \alpha \wedge \beta \wedge \varphi \wedge \psi\})$. We have to prove that $\mathcal{M}_s \preceq \mathcal{M}_{s+1}$, which is by definition equivalent to $\mathcal{M}_s \supseteq \mathcal{M}_{s+1}$. Take a valuation $m \in \mathcal{M}_{s+1}$, then $m \models Th(\mathcal{M}_s)$, and as $\mathcal{M}_s$ is closed, we have $m \in \mathcal{M}_s$.
   • *eager:* Again take an $\mathcal{M} \in \mathcal{T}_S(X)$, then again the characterization result holds. Now suppose for some $s \in \mathbb{N}$ we have $\mathcal{M}_s = \mathcal{M}_{s+1}$. We will prove that the same rules are applicable in $\mathcal{M}$ at time s, as at time $s+1$. So let us take any rule (from S), of the form $\alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma)$, where, as before, α is a conjunction of formulae of the form $H_0(K\epsilon)$, β a conjunction of formulae of the form $\neg H_0(K\delta)$, φ a conjunction of formulae of the form $\neg F(K\theta)$, and finally ψ is a conjunction of formulae of the form $K\zeta$. It is easy to see that $(\mathcal{M}, s) \models \alpha \wedge \beta \Leftrightarrow (\mathcal{M}, s+1) \models \alpha \wedge \beta$ since it only depends on $\mathcal{M}_0$. As $\mathcal{M}_s = \mathcal{M}_{s+1}$, it is straightforward that $(\mathcal{M}, s) \models \psi \Leftrightarrow (\mathcal{M}, s+1) \models \psi$. As we are dealing with closed models, whether $(\mathcal{M}, s) \models \varphi$ only depends on whether $\lim \mathcal{M} \not\models K\theta$ for all conjuncts $\neg F(K\theta)$ in ψ, and the same holds for time point $s+1$. This means that $\mathcal{M}_{s+2} = Mod(Th(\mathcal{M}_{s+1} \cup \{\gamma \mid \text{there is a rule } \alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma) \text{ in } S, \text{ applicable in } \mathcal{M} \text{ at time point } s+1\}) = Mod(Th(\mathcal{M}_s \cup \{\gamma \mid \text{there is a rule } \alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma) \text{ in } S, \text{ applicable in } \mathcal{M} \text{ at time point } s\}) = \mathcal{M}_{s+1}$. Using induction, this argument can be used to show that $\mathcal{M}_t = \mathcal{M}_s$ for all $t > s$, showing eagerness.
   • *non-inclusiveness and uniqueness of traces:* Take $\mathcal{M}, \mathcal{N} \in \mathcal{T}_S(X)$ and suppose that $\lim \mathcal{M} \preceq \lim \mathcal{N}$. We will prove that $\mathcal{N}_t \preceq \mathcal{M}_t$ for all $t \in \mathbb{N}$, using induction. Since $\mathcal{M}, \mathcal{N} \in \mathcal{T}_S(X)$, we have by definition that $Th(\mathcal{M}_0) = Cn(X) = Th(\mathcal{N}_0)$, and as both are closed models, we have $\mathcal{M}_0 = \mathcal{N}_0$. Now suppose that for some $s \in \mathbb{N}$, it holds that $\mathcal{N}_i \preceq \mathcal{M}_i$ for all $i \leq s$. Take a rule $\alpha \wedge \beta \wedge \varphi \wedge \psi \rightarrow X(K\gamma)$ of S as above, and suppose $(\mathcal{N}, s) \models \alpha \wedge \beta \wedge \varphi \wedge \psi$. As $\mathcal{N}_0 = \mathcal{M}_0$, we have $(\mathcal{M}, s) \models \alpha \wedge \beta$. As $\mathcal{N}_s \preceq \mathcal{M}_s$, we have $(\mathcal{M}, s) \models \psi$. Now take a formula $\neg F(K\theta)$ from φ. Since $(\mathcal{N}, s) \models \neg F(K\theta)$, we have (as $\mathcal{N}$ is closed), $\lim \mathcal{N} \not\models K\theta$, but as $\lim \mathcal{M} \preceq \lim \mathcal{N}$ by assumption, we

have lim $\mathcal{M} \not\models K\theta$, whence $(\mathcal{M}, s) \models \neg F(K\theta)$. This means that $(\mathcal{M}, s) \models \alpha \wedge \beta \wedge \varphi \wedge \psi$. So all rules applicable in $\mathcal{N}$ at time s, are also applicable in $\mathcal{M}$ at time s. Furthermore, as $\mathcal{N}_s \preceq \mathcal{M}_s$, we have $Th(\mathcal{N}_s) \subseteq Th(\mathcal{M}_s)$. As both are minimal models, and therefore satisfy the characterization above, we must have that $Th(\mathcal{N}_{s+1}) \subseteq Th(\mathcal{M}_{s+1})$, which means that $\mathcal{N}_{s+1} \preceq \mathcal{M}_{s+1}$. This, however, together with the fact that $\mathcal{N}_0 = \mathcal{M}_0$, means that $\mathcal{N} \preceq_0^g \mathcal{M}$. But since $\mathcal{M}$ is a $\preceq_0^g$-minimal model of S (and so is $\mathcal{N}$), this must mean that $\mathcal{M} = \mathcal{N}$. Of course this also implies that lim $\mathcal{M}$ = lim $\mathcal{N}$. As also lim $\mathcal{M}$ = lim $\mathcal{N} \Rightarrow$ lim $\mathcal{M} \preceq$ lim $\mathcal{N}$, we have proved both non-inclusiveness and uniqueness of traces.

• *invariance:* We have to prove that, for any set X, it holds that $\mathcal{T}_S(X) = \mathcal{T}_S(Cn(X))$. Now $\mathcal{T}_S(X)$ collects all minimal models $\mathcal{M}$ of S for which $Th(\mathcal{M}_0) = Cn(X)$, and $\mathcal{T}_S(Cn(X))$ collects all minimal models for which $Th(\mathcal{M}_0) = Cn(Cn(X))$, which is obviously equal as $Cn(X) = Cn(Cn(X))$.

2. Suppose $\mathcal{R}$ is a reasoning frame operator satisfying all of the above mentioned properties. We will construct an infinitary theory of reasoning S such that $\mathcal{T}_S = \mathcal{R}$. This theory S will be the union of theories S_X, one for each possible input set $X \subseteq \mathcal{L}$. In fact, as $\mathcal{R}$ satisfies invariance, whenever $Cn(X) = Cn(Y)$, we will take $S_X = S_Y$. The formulae in S_X are meant to be applicable only for traces in $\mathcal{R}(X)$, therefore such a trace must start with (exactly) the information in X. Formulae in S_X will have the following formula $\alpha_X \wedge \beta_X$ as part of their left hand side:

$$\bigwedge\{H_0(K\epsilon) \mid \epsilon \in Cn(X)\} \wedge \bigwedge\{\neg H_0(K\delta) \mid \delta \notin Cn(X)\}.$$

In words: all formulae from $Cn(X)$ must be known in the initial state, but no formula outside of $Cn(X)$ must be known. As an aside, it is not necessary to take *all* formulae from $Cn(X)$: all formulae from X is sufficient (for the left part), or even all formulae from a set of generators A (where A is a set of generators for $Cn(X)$ if $Cn(A) = Cn(X)$). Now let us consider $\mathcal{R}(X)$. If $\mathcal{R}(X) = \emptyset$, then we must make sure that there are no minimal models of S starting with X. So in this case we let S_X contain a single rule

$$\alpha_X \wedge \beta_X \to X(K\bot).$$

Let us now consider the situation that $\mathcal{R}(X) \neq \emptyset$. In this case S_X will be the union of theories $S_{X,\mathcal{M}}$ for each $\mathcal{M} \in \mathcal{R}(X)$. The rules in a theory $S_{X,\mathcal{M}}$ should be applicable only in $\mathcal{M}$, so we need part of the precondition to distinguish between the traces in $\mathcal{R}(X)$, and we can use a conjunction of formulae of the form $\neg F(K\theta)$ for this purpose. As $\mathcal{R}$ satisfies uniqueness of traces, we know that two different traces in $\mathcal{R}(X)$ have two different limits. Thus we can distinguish between traces on the basis of the limit. Furthermore, as $\mathcal{R}$ satisfies non-inclusiveness, we know that these limits are not in the relation $\preceq$, which means that they are not subsets of each other. As the traces in $\mathcal{R}(X)$

are closed (remember that this was an additional property of the information state frame), the limits are closed too. The relevance of this is that if for two closed information states M, N, we have that $M \not\subseteq N$, then there exists a propositional formula φ such that $N \models K\varphi$ but $M \not\models K\varphi$. For any two different traces $\mathcal{M}, \mathcal{N}$ in $\mathcal{R}(X)$, let $\theta_{\mathcal{M},\mathcal{N}}$ be any propositional formula such that $\lim \mathcal{N} \models K(\theta_{\mathcal{M},\mathcal{N}})$, but $\lim \mathcal{M} \not\models K(\theta_{\mathcal{M},\mathcal{N}})$. Then define formulae $\varphi_{X,\mathcal{M}}$ as follows:

$$\varphi_{X,\mathcal{M}} = \bigwedge\{\neg FK(\theta_{\mathcal{M},\mathcal{N}}) \mid \mathcal{N} \in \mathcal{R}(X), \mathcal{N} \neq \mathcal{M}\}.$$

It can easily be verified (using closedness of the traces) that $\varphi_{X,\mathcal{M}}$ is only applicable in $\mathcal{M}$ (for every other trace $\mathcal{N} \in \mathcal{R}(X)$, the formula $FK(\theta_{\mathcal{M},\mathcal{N}})$ is true, making the disjunction false). If $\mathcal{R}(X)$ contains just one trace, we may set $\varphi_{X,\mathcal{M}} = \top$ (or leave it out altogether). By using the conjunction $\alpha_X \wedge \beta_X \wedge \varphi_{X,\mathcal{M}}$ on the left hand side, we can make implications which are applicable only on $\mathcal{M}$. Now we have to make sure that the information of $\mathcal{M}$ is added at the right moments. We will have to use the formulae of the form $K\zeta$ on the left hand side for this. Now consider a part of $\mathcal{M}$ consisting of three states, $\mathcal{M}_s, \mathcal{M}_{s+1}, \mathcal{M}_{s+2}$. Then the formulae newly known in $\mathcal{M}_{s+2}$, i.e. $Th(\mathcal{M}_{s+2}) \setminus Th(\mathcal{M}_{s+1})$ can be added after state $\mathcal{M}_{s+1}$. So how can we express that we are in $\mathcal{M}_{s+1}$? Well, $\mathcal{M}_{s+1}$ is the first state where we know the formulae from $Th(\mathcal{M}_{s+1}) \setminus Th(\mathcal{M}_s)$! This means we can use any formula from that set as a precondition (preceded by a K operator) for adding the (new) conclusions for $\mathcal{M}_{s+1}$. This will now be formalized.

Let $k_{\mathcal{M}}$ be the index where the trace $\mathcal{M}$ is no longer strictly increasing (if there is such a point). Then eagerness ensures that after that index, the trace is constant. Define

$$k_{\mathcal{M}} = \begin{cases} \min\{s \mid \mathcal{M}_s = \mathcal{M}_{s+1}\} & \text{if there exists } s \text{ with } \mathcal{M}_s = \mathcal{M}_{s+1} \\ \infty & \text{otherwise} \end{cases}$$

For $0 < s < k_{\mathcal{M}}$, let $\zeta_{s,\mathcal{M}}$ be a propositional formula in $Th(\mathcal{M}_s) \setminus Th(\mathcal{M}_{s-1})$. Furthermore, let $\zeta_{0,\mathcal{M}}$ be any formula in $Th(\mathcal{M}_0)$. We are now ready to define the set of rules for the trace $\mathcal{M}$:

$$\begin{aligned} S_{X,\mathcal{M}} \quad = \quad & \{\alpha_X \wedge \beta_X \wedge \varphi_{X,\mathcal{M}} \wedge K(\zeta_{s,\mathcal{M}}) \to X(K\gamma) \mid \\ & \gamma \in Th(\mathcal{M}_{s+1}) \setminus Th(\mathcal{M}_s), 0 \leq s \leq k_{\mathcal{M}}\}. \end{aligned}$$

As announced earlier, we set

$$S_X = \bigcup_{\mathcal{M} \in \mathcal{R}(X)} S_{X,\mathcal{M}}$$

and finally

$$S = \bigcup_{X \subseteq \mathcal{L}} S_X$$

where this last union can be taken over deductively closed sets X only.

It now remains to prove that $\mathcal{T}_S = \mathcal{R}$. Take any $X \subseteq \mathcal{L}$. We will first show that $\mathcal{R}(X) \subseteq \mathcal{T}_S$. So let $\mathcal{M} \in \mathcal{R}(X)$. Then it easy to see that rules from S_Y where $Cn(Y) \neq Cn(X)$ are not applicable (at any time point) in $\mathcal{M}$. Concentrating on S_X, None of the rules in $S_{X,\mathcal{N}}$ for any $\mathcal{N} \neq \mathcal{M}$, $\mathcal{N} \in \mathcal{R}(X)$ are applicable anywhere in $\mathcal{M}$, since $(\mathcal{M}, s) \models FK(\theta_{\mathcal{N},\mathcal{M}})$, whereas the negation of this formula is part of the left hand side of every formula in $S_{X,\mathcal{N}}$. Turning to the formulae in $S_{X,\mathcal{M}}$, the part $\alpha_X \wedge \beta_X \wedge \varphi_{X,\mathcal{M}}$ is true in $\mathcal{M}$ at all points in time. As for the part $\zeta_{s,\mathcal{M}}$, this part is true in $\mathcal{M}$ at time point s and later, but false earlier. Thus we see that the rules first applicable in $\mathcal{M}$ at time point s are exactly the rules

$$\{\alpha_X \wedge \beta_X \wedge \varphi_{X,\mathcal{M}} \wedge K(\zeta_{s,\mathcal{M}}) \to X(K\gamma) \mid \gamma \in Th(\mathcal{M}_{s+1} \setminus Th(\mathcal{M}_s)\}.$$

The conclusions of other applicable rules are already in $Th(\mathcal{M}_s)$. Furthermore, $Mod(Th(\mathcal{M}_s) \cup \{\gamma \mid \gamma \in Th(\mathcal{M}_{s+1} \setminus Th(\mathcal{M}_s)\}) = \mathcal{M}_{s+1}$ (as $\mathcal{M}$ is closed and conservative). But that means that $\mathcal{M}$ is a $\preceq_0^g$-minimal model of S by the characterization result with which we started the proof of this theorem. And as $Th(\mathcal{M}_0) = Cn(X)$, we have $\mathcal{M} \in \mathcal{T}_S(X)$.

For the converse, suppose $\mathcal{M} \in \mathcal{T}_S(X)$. First of all, suppose no rule from S is applicable in $\mathcal{M}$. Using the characterization result of the start of this proof (and the fact that, by definition, $\mathcal{T}_S(X)$ contains $\preceq_0^g$-minimal models of S), this means that $\mathcal{M}_s = Mod(X)$ for all s. If $\mathcal{R}(X)$ contains any trace $\mathcal{N}$, then from the first part of the proof we have that $\mathcal{N} \in \mathcal{T}_S(X)$. We then have $\mathcal{M} \preceq_0^g \mathcal{N}$: $\mathcal{M}_0 = \mathcal{N}_0$ (as both are in $\mathcal{T}_S(X)$), and $\mathcal{M}_s = \mathcal{M}_0 = \mathcal{N}_0 \preceq \mathcal{N}_s$ (using conservativity). As $\mathcal{N}$ is a minimal model of S (being a member of $\mathcal{T}_S(X)$) this must mean that $\mathcal{N} = \mathcal{M}$ whence $\mathcal{M} \in \mathcal{R}(X)$. On the other hand, $\mathcal{R}(X)$ cannot be empty, since in that case S would contain a rule $\alpha_X \wedge \beta_X \to X(K\bot)$, which is obviously false in $\mathcal{M}$.

Now suppose that there is a rule from S applicable in $\mathcal{M}$ at time point s. Then this rule must be of the form

$$\alpha_X \wedge \beta_X \wedge \varphi_{X,\mathcal{N}} \wedge K\zeta_{t,\mathcal{N}} \to X(K\gamma) \tag{7.1}$$

for some $\mathcal{N} \in \mathcal{R}(X)$, $\zeta_{t,\mathcal{N}} \in \mathcal{N}_t \setminus \mathcal{N}_{t-1}$ (or if $t = 0$, we have $\zeta_{t,\mathcal{N}} \in \mathcal{N}_0$), and $\gamma \in Th(\mathcal{N}_{t+1}) \setminus Th(\mathcal{N}_t)$ for some $t \leq s$. We will prove that $\mathcal{N} \preceq_0^g \mathcal{M}$ using induction. First of all, as $(\mathcal{M}, s) \models \alpha_X \wedge \beta_X$, we have $Th(\mathcal{M}_0) = Cn(X)$, and as $\mathcal{N} \in \mathcal{R}(X)$, we also have $Th(\mathcal{N}_0) = Cn(X)$. Closedness of these two traces implies that $\mathcal{M}_0 = \mathcal{N}_0$. Let $u \in \mathbb{N}$ be such that $\mathcal{N}_x \preceq \mathcal{M}_x$ for all $x \leq u$. We will prove that $\mathcal{N}_{x+1} \preceq \mathcal{M}_{x+1}$. As $\mathcal{N} \in \mathcal{R}(X)$ and $\mathcal{R}(X) \subseteq \mathcal{T}_S(X)$ (first part of the proof), we know that both $\mathcal{N}$ and $\mathcal{M}$ are $\preceq_0^g$-minimal models of S. This means that both fit the characterization at the beginning of the proof, which implies (as $\mathcal{N}_u \preceq \mathcal{M}_u$) that we only have to prove that all rules applicable in $\mathcal{N}$ at time point u are also applicable in $\mathcal{M}$ at time point u. The rules applicable in $\mathcal{N}$ at time point u are exactly the rules

$$\alpha_X \wedge \beta_X \wedge \varphi_{X,\mathcal{N}} \wedge \zeta_{v,\mathcal{N}} \rightarrow X(K\gamma)$$

where $v \leq u$. However, since the truth of $\alpha_X \wedge \beta_X$ only depends on $\mathcal{M}_0$, and the truth of $\varphi_{X,\mathcal{N}}$ only depends on lim $\mathcal{M}$, the fact that formula 7.1 is true in $\mathcal{M}$ at time point s, means that $(\mathcal{M}, u) \models \alpha_X \wedge \beta_X \wedge \varphi_{X,\mathcal{N}}$. Furthermore, as $\mathcal{N} \preceq \mathcal{M}$, and $(\mathcal{N}, u) \models \zeta_{v,\mathcal{N}}$, we also have $(\mathcal{M}, u) \models \zeta_{v,\mathcal{N}}$, which means that this rule is applicable in $\mathcal{M}$ at time point u. This proves that $\mathcal{N}_{u+1} \preceq \mathcal{M}_{u+1}$. By induction we now get that $\mathcal{N} \preceq_0^g \mathcal{M}$, but as both $\mathcal{N}$ and $\mathcal{M}$ are $\preceq_0^g$-minimal models of S, it must be the case that actually $\mathcal{M} = \mathcal{N}$, which means that $\mathcal{M} \in \mathcal{R}(X)$, which had to be proven.

□

The theorem shows that infinitary theories of reasoning are in a sense 'expressively complete' for the class of these well-behaved reasoning frame operators. For other temporal specification languages, other expressiveness results will be the case. As an example, if we allow a formula in an infinitary theory of reasoning to contain, as part of its premise, a conjunction of formulae of the form $F(K\alpha)$, and we take the ordering $\preceq_0^{gel}$ (which compares two models according to $\preceq^{gel}$, but only if their initial states are equal), then this language is expressively complete for the class of reasoning frame operators which satisfy all the properties mentioned above, except non-inclusiveness. The proof of this fact is analogous to the above proof and is therefore skipped. In the next section we will look at the expressiveness of a variant of default logic.

7.2 Infinitary default logic

In this section, the expressiveness of an infinitary variant of default logic, IDL, will be studied. It was shown in Section 3.1 that default logic (or actually, sets of defaults) can be seen as a specification language for reasoning frame operators. For a set of defaults D, we defined

$$\mathsf{Tr}_D(X) = \{\mathsf{Tr}(E) \mid E \in \mathsf{Ext}(D, X)\}.$$

It was also shown earlier (Example 2.12) that a set of defaults D specifies a multiple belief state operator $\mathcal{B}_D^{dl}$ given by

$$\mathcal{B}_D^{dl}(X) = \mathsf{Ext}(D, X).$$

Analogously to the previous section, the representability questions would be:

- Given a reasoning frame operator $\mathcal{R}$, does there exist a set of defaults D such that $\mathcal{R} = \mathsf{Tr}_D$?

- Given a multiple belief state operator Γ, does there exist a set of defaults D such that $\Gamma = \mathcal{B}_D^{dl}$?

We will not solve these questions here, but instead look at a (arguably easier) different representability question. A set of defaults specifies a reasoning frame operator. But a default theory, that is, a set of defaults together with a set of formulae, specifies a set of (deductively closed) theories, namely the set of its extensions. Likewise, it specifies a set of traces, namely the traces of its extensions. In a sense, this means that we are keeping the initial facts fixed. So we will investigate the expressiveness of (a variant of) default logic as a specification language for these semantical objects. In the next subsection we will introduce terminology for a certain class of sets of theories and sets of traces, and we will introduce infinitary default logic.

7.2.1 Preliminaries

In this section, by $\mathcal{L}$ we denote a language of propositional logic with a denumerable set of atoms P. These atoms will be denoted by $p, q, r, ...$ with or without subscripts. In this section, by a *theory* we *always* mean a subset of $\mathcal{L}$ *closed under propositional provability.* We will often refer to a theory as a *belief set.* For formulae $\varphi_1, \varphi_2, ..., \varphi_n$ we introduce the notation $\langle \varphi_1, \varphi_2, ..., \varphi_n \rangle$ as an abbreviation of $Cn(\{\varphi_1, \varphi_2, ..., \varphi_n\})$.

As already mentioned, we are interested in sets of theories. Of particular interest are sets of theories that form anti-chains (no belief set is a proper subset of another).

Definition 7.4 (Belief frame) A *belief frame* is a collection of belief sets (theories) such that no belief set is a proper subset of another.

Throughout this subsection we will use a running example to illustrate the ideas and constructions.

Example 7.5 (Running example) Define the following theories:

$$\begin{array}{rcl} T_1 & = & \langle p, s, t \rangle, \\ T_2 & = & \langle p, s, \neg u \rangle, \\ T_3 & = & \langle p, \neg r, \neg q, t \rangle, \text{ and} \\ T_4 & = & \langle p, \neg r, \neg q, \neg u \rangle. \end{array}$$

It is easy to see that $\mathcal{B} = \{T_1, T_2, T_3, T_4\}$ is a belief frame.

Also, we are interested in sets of traces. Throughout this subsection we will use the following notational convention. If an upper case symbol, say E, stands for a sequence of theories, then the elements of the sequence will be referred to as $E_1, E_2, \ldots$, and their union, $\bigcup_{i=1}^{\infty} E_i$, will be denoted by E^∞. Note that such a sequence forms a trace in the information state frame IS^{syn} of syntactic states

(Definition 2.6), and that E^∞ is the same as the limit of E. From here on, we will be working in IS^{syn}.

Definition 7.6 (Reasoning frame) A collection $\mathcal{T}$ of eager and conservative reasoning traces from IS^{syn} is called a *reasoning frame* if for every $T, S \in \mathcal{T}$:

1. $T_0 = S_0$, and
2. If $T^\infty \subseteq S^\infty$, then $T = S$.

It is easy to see that the limit of a conservative and eager reasoning trace is a theory, that is, it is closed under propositional provability, and that the limits of reasoning traces in a reasoning frame form a belief frame, that is, form an antichain. Furthermore, if we take a reasoning frame operator $\mathcal{R}$ in IS^{syn} consisting of eager and conservative traces, which in addition satisfies non-inclusiveness and uniqueness of traces, then for every $X \subseteq \mathcal{L}$, we have that $\mathcal{R}(X)$ is a reasoning frame. So reasoning frames are generated by fixing the initial facts for a reasoning frame operator satisfying some properties. In the same vein, belief frames result from keeping the initial facts fixed for a multiple belief state operator (in IS^{syn}) satisfying non-inclusiveness. Just as there were connections between reasoning frame operators and multiple belief state operators, one can associate a belief frame to each reasoning frame.

Definition 7.7 (Belief frame of a reasoning frame) Let $\mathcal{T}$ be a reasoning frame. The belief frame $\mathcal{B}_\mathcal{T}$ associated with $\mathcal{T}$ is defined by:

$$\mathcal{B}_\mathcal{T} = \{T^\infty : T \in \mathcal{T}\}.$$

Example 7.8 The following is an example of a reasoning frame:

$$\begin{aligned}
\mathcal{F} \quad = \quad & \{(\langle p\rangle, \langle p, s\rangle, \langle p, s, t\rangle, \langle p, s, t\rangle, \ldots), \\
& (\langle p\rangle, \langle p, s\rangle, \langle p, s, \neg u\rangle, \langle p, s, \neg u\rangle, \ldots), \\
& (\langle p\rangle, \langle p, \neg r, \neg q\rangle, \langle p, \neg r, \neg q, t\rangle, \langle p, \neg r, \neg q, t\rangle, \ldots), \\
& (\langle p\rangle, \langle p, \neg r, \neg q\rangle, \langle p, \neg r, \neg q, \neg u\rangle, \langle p, \neg r, \neg q, \neg u\rangle, \ldots)\}.
\end{aligned}$$

It is easy to show that this is indeed a reasoning frame. The reader can check that $\mathcal{B}_\mathcal{F} = \mathcal{B}$ where $\mathcal{B}$ was defined in Example 7.5.

In this section, we intend to show that infinitary default logic can specify belief frames and reasoning frames (using default *theories*, not *sets of defaults* as was done earlier), but more importantly, that any belief frame and any reasoning frame can be specified by an IDL-theory. Some results in this direction were already obtained in [MTT97], where the problem of encoding belief frames by *finitary* default theories was studied in detail. In addition to a number of positive results, it is proved in [MTT97] that not every belief frame can be represented as the family of all extensions of a default theory. In this section we will generalize default logic by allowing infinite sets of justifications. Then we will prove that infinitary default logic is powerful enough to serve as a specification language for arbitrary belief and reasoning frames. We will now proceed by defining infinitary default logic.

An *infinitary default* (*IDL-default*, for short) is an expression d:

$$d = \frac{\alpha : \Gamma}{\beta}, \tag{7.2}$$

where α and β are formulae from $\mathcal{L}$, and Γ is a set, possibly infinite, of formulae from $\mathcal{L}$. The intended reading of this expression is analogous to the finitary case:

> *if you believe α, and all formulae in Γ are consistent with what you believe, then you can conclude β.*

The formula α is called the *prerequisite* of d ($p(d)$, in symbols) and β is called the *consequent* of d ($c(d)$, in symbols). The set of formulae Γ is called the *justification* set of d and is denoted by $j(d)$. If $p(d)$ is a tautology, d is called *prerequisite-free*. In such a case, $p(d)$ is usually omitted from the notation of d. This terminology is naturally extended to a set of defaults D. Namely, the *prerequisite*, *consequent* and *justification* sets of D, in symbols $p(D)$, $c(D)$ and $j(D)$, are defined by:

$$p(D) = \bigcup_{d \in D} \{p(d)\}, \quad c(D) = \bigcup_{d \in D} \{c(d)\}, \quad j(D) = \bigcup_{d \in D} j(d).$$

A pair (D, W), where D is a set of IDL-defaults and $W \subseteq \mathcal{L}$ is a set of formulae, is called an *infinitary default theory* (or *IDT*). Rules with infinite sets of justifications were considered in [Fer91] in the context of logic programs.

We will now generalize the notion of an extension, introduced by Reiter [Rei80b] for standard default theories, to the case of IDTs. To this end, we will introduce the concept of an S-trace. This notion is closely related to the fixpoint construction of extensions presented by Reiter [Rei80b].

Definition 7.9 Let (D, W) be an IDT. Let $S \subseteq \mathcal{L}$ be a theory. By the *S-trace* of (D, W) we mean the sequence E of theories defined recursively as follows:

1. $E_0 = Cn(W)$,
2. for every integer $n \geq 0$:

$$E_{n+1} = Cn(E_n \cup \{c(d) \colon\ d \in D,\ \ p(d) \in E_n \text{ and for all } \gamma \in j(d),\ \neg\gamma \notin S\}).$$

The notion of an S-trace allows us to introduce the notion of an IDL-extension of an IDT.

Definition 7.10 Let (D, W) be an IDT. A set $S \subseteq \mathcal{L}$ is an *IDL-extension* of (D, W) if

$$S = E^{\infty},$$

where E is the S-trace for (D, W).

Clearly, each standard (finitary) default theory (with each default having only finitely many justifications) is, in particular, an IDT. Moreover, it is straightforward to check that if an IDT happens to be finitary, then the notion of an IDL-extension coincides with that of extension. Therefore, throughout this subsection we will refer to the IDL-extensions simply as extensions.

We will denote by $\mathsf{Ext}(D, W)$ the collections of all extensions of an IDT (D, W). The collection of all S-traces of (D, W), where $S \in \mathsf{Ext}(D, W)$ will be denoted by $tr(D, W)$. Two IDTs Δ, Δ' are called *limit-equivalent* if $\mathsf{Ext}(\Delta) = \mathsf{Ext}(\Delta')$.

There are several alternative characterizations of extensions of standard default theories (see e.g., [MT93]). We will now generalize one of them to the case of infinitary default theories. It can be stated in terms of the reduct of the set of defaults. A default d (a set of defaults D) is *applicable* with respect to a theory S (is S*-applicable*) if $S \not\vdash \neg\gamma$ for every $\gamma \in j(d)$ ($j(D)$, respectively). By the *reduct* D_S of D with respect to S we mean the set of monotone inference rules:

$$D_S = \left\{ \frac{\alpha}{\beta} \colon \text{ for some } \Gamma \subseteq \mathcal{L},\ \frac{\alpha : \Gamma}{\beta} \in D,\ \text{and } \frac{\alpha : \Gamma}{\beta} \text{ is } S\text{-applicable} \right\}.$$

Each set B of standard monotone inference rules determines a formal proof system, denoted by $PC + B$, in which derivations are built by means of propositional provability and rules in B. The corresponding provability operator will be denoted by $\vdash_B$ and the consequence operator by $Cn^B(\cdot)$ ([MT93]). In particular, each set D_S determines the provability operator $\vdash_{D_S}$ and the consequence operator $Cn^{D_S}(\cdot)$.

Proposition 7.11 Let D be a set of IDL-defaults, and let W and S be subsets of $\mathcal{L}$. Then, S is an extension of (D, W) if and only if $S = Cn^{D_S}(W)$.

Proof: Completely analogous to the proof of this property for Reiter default logic in [MT93]. □

Let us introduce one more useful notion. A default d is *generating* for a theory S if $p(d) \in S$ and $S \not\vdash \neg\gamma$ for every $\gamma \in j(d)$. The set of all defaults from D which are generating for S is denoted by $GD(D, S)$.

Once the reduct is computed the distinction between infinitary and standard defaults disappears. This explains why many of the properties of default logic remain true in the infinitary case. In particular, we have the following results.

Proposition 7.12 Let (D, W) be an IDT. Then:

1. If S is an extension of (D, W), then S is a belief set (theory).
2. The operator $Cn^{D_S}(W)$ is monotone in D and W, and antimonotone in S.
3. The collection $\mathsf{Ext}(D, W)$ is a belief frame. That is, if $T_1, T_2 \in \mathsf{Ext}(D, W)$ and $T_1 \subseteq T_2$, then $T_1 = T_2$.
4. If S is an extension of (D, W) then $S = Cn(W \cup c(GD(D, S)))$.
5. If all defaults in D are prerequisite-free then S is an extension of (D, W) if and only if $S = Cn(W \cup c(GD(D, S)))$.

Parts (1) and (3) of Proposition 7.12 show that IDTs can be used to represent belief frames. The next result shows that they can also be used to represent reasoning frames.

Proposition 7.13 Let (D, W) be an IDT.

1. Let S be a theory in $\mathcal{L}$. If E is the S-trace for (D, W) then E is an eager and conservative reasoning trace.
2. The collection of reasoning traces $tr(D, W)$ is a reasoning frame.

The proofs of both of the above propositions are again analogous to the proofs for the finitary case (see [MT93]) and are therefore omitted.

We can now formally introduce the notions of representability of belief frames and reasoning frames by default theories.

Definition 7.14 Let $\mathcal{B}$ be a family of belief sets contained in $\mathcal{L}$. The family $\mathcal{B}$ is *representable* by an IDT Δ if $\mathsf{Ext}(\Delta) = \mathcal{B}$. Similarly, if $\mathcal{T}$ is a family of reasoning traces, then it is *representable* by an IDT Δ if $tr(\Delta) = \mathcal{T}$.

Example 7.15 It turns out that the belief frame $\mathcal{B}$ of Example 7.5 is representable. Define the IDT (D, W) by

$$\begin{aligned} W &= \{p\}, \text{ and} \\ D &= \{\frac{p\colon q \vee r}{s}, \frac{p\colon \neg s}{\neg r}, \frac{:\neg s}{\neg q}, \frac{s \vee \neg r : u}{t}, \frac{s \vee \neg q \colon \neg t}{\neg u}\}. \end{aligned}$$

It can be easily verified that $\mathsf{Ext}(D, W) = \mathcal{B}$. Furthermore, it can also be checked that $tr(D, W) = \mathcal{F}$, where $\mathcal{F}$ was defined in Example 7.8. This means that $\mathcal{F}$ is also representable. Of course, since this IDT is finitary, $\mathcal{B}$ and $\mathcal{F}$ are also representable in standard default logic.

The notion of representability by default theories was studied in [MTT97]. A complete description of families of theories that are representable by default theories with a finite set of defaults was given there. However, the general question of representability by arbitrary default theories has not been settled yet. The main difference between a standard and an infinitary default is that the latter can encode an infinite set of constraints determining its applicability (in the form of infinite sets of justifications). Our results in the next section show that the infinitary default logic is more expressive than the default logic by Reiter. In particular, we show that every family of theories satisfying the necessary condition for representability, described in Proposition 7.12(3), is representable by an infinitary default theory.

7.2.2 Representability of belief frames by IDTs

We start with the result that allows us to replace any IDT with a limit-equivalent IDT in which all defaults are prerequisite-free.

Theorem 7.16 For every IDT Δ, there is a prerequisite-free IDT Δ', limit-equivalent to Δ.

Proof: Let $\Delta = (D, W)$. By a *quasi-proof* from D and W we mean any proof from W in the system $PC + D^m$, where

$$D^m = \left\{ \frac{\alpha}{\beta} \colon \text{ for some } \Gamma \subseteq \mathcal{L},\ \frac{\alpha : \Gamma}{\beta} \in D \right\}.$$

For every quasi-proof ϵ from D and W, let D_ϵ be the set of all defaults used in ϵ. For each such proof ϵ, define

$$d_\epsilon = \frac{: j(D_\epsilon)}{\bigwedge c(D_\epsilon)}$$

(observe that D_ϵ is finite and, so, d_ϵ is well-defined). Next, define

$$Q = \{d_\epsilon \colon \epsilon \text{ is a quasi-proof from } W\}.$$

Each default in Q is prerequisite-free. Put $\Delta' = (Q, W)$. We will show that Δ' has exactly the same extensions as (D, W). To this end, we will show that for every theory S and for every formula φ,

$$W \vdash_{D_S} \varphi \ \text{ iff } \ W \vdash_{Q_S} \varphi.$$

Assume first that $W \vdash_{D_S} \varphi$. Then, there is a quasi-proof ϵ of φ such that all defaults in D_ϵ are applicable with respect to S. Moreover, $W \cup c(D_\epsilon) \vdash \varphi$. Observe that $c(d_\epsilon) \vdash c(D_\epsilon)$. Since d_ϵ is prerequisite-free and S-applicable, $W \vdash_{Q_S} W \cup c(D_\epsilon)$. Hence, $W \vdash_{Q_S} \varphi$.

To prove the converse implication, observe that since all defaults in Q are prerequisite-free,

$$\{\varphi \colon W \vdash_{Q_S} \varphi\} = Cn(W \cup c(Q_S)).$$

Hence, it is enough to show that

$$W \vdash_{D_S} W \cup c(Q_S).$$

Clearly, for every $\varphi \in W$, $W \vdash_{D_S} \varphi$. Consider $\varphi \in c(Q_S)$. It follows that there is a quasi-proof ϵ such that d_ϵ is S-applicable and $c(d_\epsilon) = \varphi$. Consequently, all defaults occurring in ϵ are S-applicable. Thus, for every default $d \in D_\epsilon$,

$$W \vdash_{D_S} c(d).$$

Since $\varphi = \bigwedge c(D_\epsilon)$,

$$W \vdash_{D_S} \varphi.$$

$\square$

Example 7.17 Let us look at the IDT (D, W) defined in Example 7.15. Every default d_ϵ defined in the proof of Theorem 7.16 is uniquely determined by the set D_ϵ of defaults used in ϵ. This means that for (D, W), which consists of five defaults, at most $2^5 = 32$ defaults d_ϵ can be defined. It turns out that the IDT defined in this way actually contains 24 defaults (the other subsets of the defaults in D can not be combined into a proof). Rather than listing all 24, we will give a number of them. First of all, the defaults with prerequisite in W are proofs, so for instance $\frac{: q \vee r}{s}$ and $\frac{: \neg s}{\neg q}$ are in Q. The second, third and fifth defaults in D give rise to the following default:

$$\frac{: \{\neg s, \neg t\}}{\neg r \wedge \neg q \wedge \neg u}.$$

But there are also defaults that contradict their own justification, such as:

$$\frac{: \{q \vee r, u, \neg t\}}{s \wedge t \wedge \neg u}.$$

These defaults are present in Q, but they are harmless given the other defaults in Q.

Proposition 7.12 implies that for every infinitary default theory (D, W), its family of extensions $\mathsf{Ext}(D, W)$ is a belief frame (cf. parts (1) and (3)). To answer the question whether the converse is true as well, by Theorem 7.16 we can concentrate on prerequisite-free IDTs. It turns out that every belief frame is representable by a (prerequisite-free) IDT.

Theorem 7.18 Let $\mathcal{B}$ be a family of belief sets. Then the following statements are equivalent:

(i) $\mathcal{B}$ is a belief frame,

(ii) $\mathcal{B}$ is representable by a prerequisite-free IDT.

Proof: It suffices to prove that any belief frame is representable by a prerequisite-free IDT. To this end, let us consider a belief frame $\mathcal{B}$. If $\mathcal{B} = \emptyset$ then take any (Reiter) default theory without extensions. If $\mathcal{B} = \{T\}$, then define $D = \emptyset$. Clearly, $\mathsf{Ext}(D,T) = \mathcal{B}$.

Hence, assume that $\mathcal{B}$ contains at least two theories. Since no theory in $\mathcal{B}$ is a proper subtheory of another, it follows that all theories contained in $\mathcal{B}$ are consistent.

For every $S, T \in \mathcal{B}$ such that $S \neq T$, define $\varphi_{S,T}$ to be any formula belonging to $S \setminus T$. For every $T \in \mathcal{B}$, define

$$D^T = \left\{ \frac{:\{\neg\varphi_{S,T} : S \in \mathcal{B},\ S \neq T\}}{\psi} :\ \psi \in T \right\}.$$

Finally, define

$$D = \bigcup_{T \in \mathcal{B}} D^T.$$

We will show that $\mathsf{Ext}(D, \emptyset) = \mathcal{B}$.

Consider $T \in \mathcal{B}$. Then $D_T = \{\frac{:}{\psi} : \psi \in T\}$. Hence, $Cn^{D_T}(\emptyset) = T$ and T is an extension of $(D, \emptyset)$.

Conversely, let T be an extension of $(D, \emptyset)$. We have just proved that $\mathcal{B} \subseteq \mathsf{Ext}(D, \emptyset)$. Consequently, $(D, \emptyset)$ has at least two extensions. It follows that $Cn(\emptyset)$ is not an extension of $(D, \emptyset)$ (the theory $Cn(\emptyset)$ is a subset of every extension of (D, W)). In particular, $T \neq Cn(\emptyset)$. Therefore, the set D_T is not empty.

Consider a set $S \in \mathcal{B}$. Observe that all defaults in D^S have the same set of justifications. Consequently, either all of them are generating for T or none. It follows that T is the union of a nonempty (since $D_T \neq \emptyset$) family of theories in $\mathcal{B}$. If T is the union of at least two theories, than $D_T = \emptyset$, a contradiction. Hence, $T = S$, for some $S \in \mathcal{B}$. That is, $T \in \mathcal{B}$. $\square$

Example 7.19 We already know that our example belief frame $\mathcal{B}$ is representable, and we know it is representable by a prerequisite-free IDT. In order to illustrate the construction process of the proof of Theorem 7.18, we will perform this for $\mathcal{B}$. Note, first of all, that in the definition of D^T, we need not add defaults for every $\varphi \in T$, but that it is sufficient to do this for a set of generators of T (T is generated by a set of formulae if it is the propositional closure of this set). Furthermore, when a formula $\varphi_{S,T}$ is a negation, we will eliminate the double negation in the default. We will now construct the sets D^T:

1. T_1: first we must choose the formulae φ_{S,T_1}. Take $\varphi_{T_2,T_1} = \neg u$, $\varphi_{T_3,T_1} = \neg r$, and $\varphi_{T_4,T_1} = \neg r$, then
$$D^{T_1} = \{\frac{:\{u,r\}}{p}, \frac{:\{u,r\}}{s}, \frac{:\{u,r\}}{t}\}.$$
Note that these defaults have the same set of justifications, so instead of taking 3 defaults, we can also form one by taking the conjunction of the consequents. We will do this for the remaining theories.

2. T_2: Let $\varphi_{T_1,T_2} = t$, $\varphi_{T_3,T_2} = t$, $\varphi_{T_4,T_2} = \neg r$, and define
$$D^{T_2} = \{\frac{:\{\neg t, r\}}{p \wedge s \wedge \neg u}\}.$$

3. T_3: Let $\varphi_{T_1,T_3} = s$, $\varphi_{T_2,T_3} = s$, $\varphi_{T_4,T_3} = \neg u$, and define
$$D^{T_3} = \{\frac{:\{\neg s, u\}}{p \wedge \neg r \wedge \neg q \wedge t}\}.$$

4. T_4: Let $\varphi_{T_1,T_4} = s$, $\varphi_{T_2,T_4} = s$, $\varphi_{T_3,T_4} = t$, and define
$$D^{T_4} = \{\frac{:\{\neg s, \neg t\}}{p \wedge \neg r \wedge \neg q \wedge \neg u}\}.$$

If we define $D = D^{T_1} \cup D^{T_2} \cup D^{T_3} \cup D^{T_4}$, then it can be checked that indeed $\mathsf{Ext}(D, \emptyset) = \mathcal{B}$.

Theorem 7.18 and the results in [MTT97] imply that infinitary default logic is a more powerful knowledge representation formalism than classical default logic. In other words, allowing infinite justification sets leads to a more expressive specification language.

Corollary 7.20 There are belief frames representable by an IDT but not representable by a standard default theory.

We will give an example. Let $\{p_0, p_1, \dots\}$ be a set of propositional atoms. Define $T_i = Cn(\{p_i\})$, $i = 0, 1, \dots$, and $\mathcal{B} = \{T_i : i = 0, 1, \dots\}$. It is clear that $\mathcal{B}$ consists of non-including theories, and is therefore representable by an IDT. If we define
$$\begin{array}{rcl} W & = & \emptyset \quad \text{and} \\ D & = & \{\frac{:\{\neg p_j \mid j \neq i\}}{p_i} \mid i \geq 0\}, \end{array}$$
then it can be easily verified that $\mathsf{Ext}(D, W) = \mathcal{B}$. It was shown, however, in [MTT97] (Theorem 3.5), that $\mathcal{B}$ is not representable by a (Reiter) default theory.

As another corollary, we obtain the result already proved in [MTT97].

Proposition 7.21 Let $\mathcal{B}$ be a finite belief frame. Then $\mathcal{B}$ is representable by a (Reiter) default theory (possibly with an infinite set of defaults).

This is actually a special case of a more general criterion for representability by (Reiter) default theories. Let us call a family of theories $\mathcal{B}$ *finitely distinguishable* if for all $T \in \mathcal{B}$ there exists a *finite* set FD(T) such that $FD(T) \cap T = \emptyset$ and $\forall S \in \mathcal{B} : S \neq T \Rightarrow FD(T) \cap S \neq \emptyset$. We have the following result.

Proposition 7.22 Every finitely distinguishable belief frame is representable by a Reiter default theory.

Proof: In the proof of Theorem 7.18, we can always choose the formulae $\varphi_{S,T}$ from $FD(T)$, a finite set. Then the sets D^T contain only defaults with finite justification sets, so that the IDT defined in the proof is in fact a Reiter default theory. □

It is easy to see that a finite belief frame is finitely distinguishable. Hence, Proposition 7.22 applies to finite belief frames.

7.2.3 Representability of reasoning frames by IDTs

In the previous section we proved that any belief frame can be represented by a prerequisite-free IDL-theory. In this section we will look not only at the outcomes of a reasoning process, a belief frame, but also at the process in which these outcomes are constructed. Note that by using prerequisites that logically depend on consequents of other defaults, it is possible to express constraints on the order in which states occur in a trace. Using this observation, we will study the question whether infinitary default logic can be used as a specification language for collections of traces — reasoning frames. The main result of this section is that every reasoning frame is representable by an IDT.

Theorem 7.23 Let $\mathcal{T}$ be a collection of reasoning traces. Then the following statements are equivalent:

(i) $\mathcal{T}$ is a reasoning frame,

(ii) $\mathcal{T}$ is representable by an IDT.

Proof: If there is an IDT Δ such that $\mathcal{T} = tr(\Delta)$, then $\mathcal{T}$ is a reasoning frame by Proposition 7.13. To prove the converse implication, we proceed as follows. If $\mathcal{T}$ is empty, we can take Δ to be any default theory without extensions. So suppose that $\mathcal{T} \neq \emptyset$. Take any trace $T \in \mathcal{T}$, and define $W = T_0$. As $\mathcal{T}$ is a reasoning frame, we have that $W = S_0$ for all traces $S \in \mathcal{T}$.

Consider a trace $T \in \mathcal{T}$. Then T is increasing, and may become constant from a certain index on. We define this index k_T by

$$k_T = \begin{cases} \min\{i\colon T_i = T_{i+1}\} & \text{if there exists an } i \text{ with } T_i = T_{i+1} \\ \infty & \text{otherwise.} \end{cases}$$

Now for $0 < i < k_T$, define $\psi_{i,T}$ to be any formula in $T_i \setminus T_{i-1}$, and define $\psi_{0,T}$ as any formula in T_0. These formulae will serve as prerequisites for defaults that will 'fire' in order to form T_{i+1}.

For the justifications of rules, we will use the same construction as used in the proof of Theorem 7.18. For any $S \in \mathcal{T}$ such that $S \neq T$, define $\varphi_{S,T}$ to be any formula belonging to $S^\infty \setminus T^\infty$. Since $\mathcal{T}$ is a reasoning frame and $S \neq T$, $S^\infty \not\subseteq T^\infty$. Hence, $\varphi_{S,T}$ can indeed be found. Now define

$$D^T = \left\{ \frac{\psi_{i,T}\colon \{\neg\varphi_{S,T}\colon S \in \mathcal{T},\ S \neq T\}}{\chi} \colon \chi \in T_{i+1} \setminus T_i,\ 0 \leq i < k_T \right\}.$$

Finally, define

$$D = \bigcup_{T \in \mathcal{T}} D^T.$$

We will show that $tr(D, W) = \mathcal{T}$.

Consider $T \in \mathcal{T}$. First observe that, by definition, $T_0 = W = Cn(W)$. Furthermore, the set of defaults in D which are applicable for T^∞ is exactly D^T. It follows that

$$\{c(d)\colon d \in D, p(d) \in T_i, d \text{ is } T^\infty\text{–applicable}\} = \{\chi\colon \chi \in T_{i+1} \setminus T_0\}.$$

As $T_0 \subseteq T_i$, we have that

$$T_{i+1} = Cn(T_i \cup \{c(d)\colon d \in D, p(d) \in T_i, d \text{ is } T^\infty\text{–applicable}\}).$$

From this we conclude that $T \in tr(D, W)$.

For the converse, suppose that $T \in tr(D, W)$. If none of the defaults in D are T^∞–applicable, then $T_i = W$ for all i. Consider an $S \in \mathcal{T}$. Then, we have $S \in tr(D, W)$. Now, since $S^\infty \supseteq W$ and extensions form an anti-chain, $S^\infty = W$. Hence, $S = T$ and $T \in \mathcal{T}$.

So suppose there is a T^∞–applicable default in D. Then there exists a trace $S \in \mathcal{T}$ such that all defaults in D^S are T^∞–applicable. We will show by induction that $S_i \subseteq T_i$. Indeed, if $i = 0$, then $S_0 = W = T_0$. For the induction step, observe that

$$T_{i+1} = Cn(T_i \cup \{c(d)\colon d \in D, p(d) \in T_i, d \text{ is } T^\infty\text{–applicable}\}) \supseteq$$

$$Cn(S_i \cup \{c(d)\colon d \in D, p(d) \in S_i, d \in D^S\})$$

Since $S \in tr(D,W)$ and D^S is exactly the set of defaults of D which are S^∞–applicable, the last term is equal to S_{i+1}.

Now we have that $S^\infty \subseteq T^\infty$. Moreover, since both S^∞ and T^∞ are extensions of (D,W), it follows that $S^\infty = T^\infty$. But then a default is S^∞–applicable if and only if it is T^∞–applicable, so that $S_i = T_i$ for all i, or $S = T$. We conclude that $T \in \mathcal{T}$. □

Notice the similarity between some of the constructions in this proof (the $\varphi_{S,T}$ and $\psi_{i,T}$) and constructions in the proof of Theorem 7.3 (the $\varphi_{X,\mathcal{M}}$ and $\zeta_{s,\mathcal{M}}$).

Example 7.24 We will give a reasoning frame such that its associated belief frame is $\mathcal{B}$ (different from the one we gave earlier), and construct an IDT that represents it. The reasoning frame consists of the following traces:

$$\begin{aligned}
T^1 &: (\langle p\rangle, \langle p, s\vee t\rangle, \langle p,s,t\rangle ...),\\
T^2 &: (\langle p\rangle, \langle p, s, \neg u\rangle ...),\\
T^3 &: (\langle p\rangle, \langle p, t\rightarrow \neg r\wedge\neg q\rangle, \langle p, t, \neg r, \neg q\rangle ...),\\
T^4 &: (\langle p\rangle, \langle p, \neg r\rangle, \langle p, \neg r, \neg q\rangle, \langle p, \neg r, \neg q, \neg u\rangle ....).
\end{aligned}$$

It can again easily be seen that this is a reasoning frame. We define $W = Cn(\{p\})$. We will take the $\varphi_{S,T}$ the same as in Example 7.19. Then we have to define the formulae $\psi_{i,T}$. We will not enumerate all of these explicitly, but just give an example. Consider ψ_{1,T^4}. This has to be a formula in $(T^4)_1 \setminus (T^4)_0 = \langle p, \neg r\rangle \setminus \langle p\rangle$, so we could take $\psi_{1,T^4} = \neg r$. We now give the sets of defaults:

$$\begin{aligned}
D^{T^1} &= \{\frac{p\colon \{u,r\}}{s\vee t}, \frac{s\vee t\colon \{u,r\}}{s\wedge t}\},\\
D^{T^2} &= \{\frac{p\colon \{\neg t,r\}}{s\wedge\neg u}\},\\
D^{T^3} &= \{\frac{p\colon \{\neg s,u\}}{t\rightarrow\neg r\wedge\neg q}, \frac{t\rightarrow\neg r\wedge\neg q\colon \{\neg s,u\}}{t}\},\\
D^{T^4} &= \{\frac{p\colon \{\neg s,\neg t\}}{\neg r}, \frac{\neg r\colon \{\neg s,\neg t\}}{\neg q}, \frac{\neg q\colon \{\neg s,\neg t\}}{\neg u}\}.
\end{aligned}$$

The reader can check that, setting $D = D^{T^1}\cup D^{T^2}\cup D^{T^3}\cup D^{T^4}$, indeed $tr(D,W) = \{T^1, T^2, T^3, T^4\}$.

As was the case in the construction of an IDL-theory in the previous section, we again have considerable freedom in choosing the formulae $\varphi_{S,T}$. A second source of freedom comes from the choice of the prerequisites in the above construction. Thus, in general there are many different theories which all specify the same reasoning frame.

From the construction in the proof it is clear that, analogous to the case of belief frames, reasoning frames with a *finite* number of reasoning traces can be represented by a (Reiter) default theory.

One could ask if finitary representability of the belief frame of a reasoning frame implies that the reasoning frame itself has a finitary representation. A cardinality argument shows that this is not the case. Specifically, let us consider the belief frame $\mathcal{B}$ consisting of all complete theories over the set of atoms $\{p_1, p_2, \ldots\}$. This belief frame has a finitary representation (see [MTT97], Corollary 5.5). It is easy to see that there are more than continuum reasoning frames with belief frame $\mathcal{B}$: there are continuum many different complete theories over these atoms, and for each complete theory, there are at least continuum many different traces leading to it (for each atom p_i we have a choice at which time point j to add it to the trace). On the other hand, there are only continuum many finitary default theories.

7.2.4 Multiple belief state operators and reasoning frame operators

In the preceding sections we have seen that infinitary default logic can be used for the specification of belief frames and reasoning frames. These two notions describe the reasoning process of an agent (on two levels of abstraction) from a *fixed* set of initial facts. As already mentioned earlier, we are also interested in the expressiveness of default logic for specifying multiple belief state operators and reasoning frame operators (which formalize reasoning from *any* set of initial facts). For every $X \subseteq \mathcal{L}$, we have that $\mathcal{B}(X)$ ($\mathcal{R}(X)$ respectively) is a belief frame (reasoning frame, respectively) if $\mathcal{B}$ is a multiple belief state operator ($\mathcal{R}$ a reasoning frame operator) in IS^{syn}. So by taking a different set of defaults for every set of inputs, such operators can be specified.

Definition 7.25

1. Let $\mathcal{B}$ be a multiple belief state operator. The operator $\mathcal{B}$ is *representable* by an indexed family of sets of defaults $(D_X)_{X \subseteq \mathcal{L}}$ if for all $X \subseteq \mathcal{L}$: $\mathcal{B}(X) = \mathsf{Ext}(D_X, X)$.
2. Let $\mathcal{F}$ be a reasoning frame operator. The operator $\mathcal{F}$ is *representable* by an indexed family of sets of defaults $(D_X)_{X \subseteq \mathcal{L}}$ if for all $X \subseteq \mathcal{L}$: $\mathcal{F}(X) = tr(D_X, X)$.

Given the results in the previous sections, the following is easy to see:

Proposition 7.26

1. A multiple belief state operator $\mathcal{B}$ is representable by an indexed family of sets of (prerequisite-free) defaults iff $\mathcal{B}(X)$ is a belief frame for all $X \subseteq \mathcal{L}$.

2. A reasoning frame operator $\mathcal{F}$ is representable by an indexed family of sets of defaults iff $\mathcal{F}(X)$ is a reasoning frame for all $X \subseteq \mathcal{L}$.

In principle, this is a valid way of specifying multiple belief state operators and reasoning trace operators. However, it is intuitively not very likely that an agent should use a (completely) different set of defaults in every situation. Instead, it seems more plausible that the agent has *one* set of defaults which (s)he uses regardless of the initial facts (meaning that $D_X = D_Y$ for all X, Y). This leads to the representability question posed in the beginning of this section: given a multiple belief state operator $\mathcal{B}$, does there exist a set of (prerequisite-free) defaults D, such that for all $X \subseteq \mathcal{L}$ we have $\mathcal{B}(X) = \mathsf{Ext}(D, X)$ (and similarly for reasoning frame operators)? It seems that this is a non-trivial question; we will leave this for future research.

7.3 Conclusions and related work

In this chapter, we investigated the expressiveness of two specification languages, infinitary theories of reasoning and infinitary default logic. It was shown in the first section that there is a natural class of reasoning frame operators such that the reasoning frame operator associated with an infinitary theory of reasoning always falls into this class, and that any member of this class can be represented by an infinitary theory of reasoning. This shows that infinitary theories of reasoning are especially suited for representing this class.

In [MTT97] the usefulness of Reiter's default logic for specifying multiple belief sets of an agent was investigated. It was established that every finite non-including family of belief sets is representable by a default theory. However, examples of denumerably infinite non-including families were constructed that are not representable by a default theory. In the second section these results have been extended in two directions. Firstly, a new variant of default logic was introduced, infinitary default logic, that allows to represent every non-including family of belief sets, independent of its cardinality.

Secondly, not only the representability of families of belief sets as an outcome of default reasoning processes was investigated, but also the representability of default reasoning traces constructing these belief sets. Here a positive answer was also obtained for infinitary default logic, whereas Reiter's default logic fails for the non-finite case.

It is interesting to note that from a representation viewpoint, the only role played by the prerequisites lies in guiding the construction process. Of course, even when specifying only belief sets, it may be the case that an IDL-theory *with* prerequisites exists which is more concise or intuitive than a prerequisite-free theory. However, this would also give a specification at a lower level of abstraction since it not only specifies the outcomes of the reasoning but also the way outcomes are generated. One can then choose to commit to this particular specification of the traces, but one could also consider the specification as meant only to specify the outcomes and give a different specification for the traces. One way of changing the specification for the traces is by introducing so-called *lemma default rules* (see e.g. [Sch92a]). This causes conclusions to be added earlier in a trace.

Further issues for research include representability of multiple belief state operators and reasoning frame operators using default logic (as mentioned in Subsection 7.2.4) and the general question of representability using finitary default logic (with infinite sets of defaults). It may be possible to adapt some of the topological methods used in [Fer94], which solves the representability question for logic programming (under some different semantics).

Acknowledgments

The material in the first section is part of [ET96c]. The material in the second section appeared earlier in [EMTT96].

Chapter 8

Applications of the Theory

In this chapter we will discuss two applications of the theory developed so far. The first section describes the use of a temporal logic to describe compositional multi-agent systems. This allows the formalization of (compositional) proofs for verification purposes. The second section describes the formalization in terms of (a slight variation of) a multiple belief state operator of a reasoning task concerning the classification of objects, and in particular the ecological classification of terrains. This formalization is the basis for an implemented system aiding ecologists.

8.1 Compositional multi-agent systems

In this section we will consider the reasoning (and acting) processes generated by multi-agent systems. A multi-agent system (MAS) consists of a number of autonomous entities (*agents*), each of which reasons about its environment (including the other agents), communicates with other agents and interacts with the external world (observation and execution of actions). We will not discuss the notion of agency any further here, but concentrate on architectures for multi-agent systems. The goal is to translate a description or specification of a multi-agent system into temporal logic. On the one hand, this can be used to give a formal semantics (a temporal semantics) to specification languages for multi-agent systems. This has been done for the language DESIRE, for example. On the other hand, temporal logic can be used to formalize verification and validation proofs for MASs, where both static and dynamic properties of such systems can be verified. This enables the construction of tools to check proofs or even automatically generate proofs. We shall start by giving a brief description of what we mean by a multi-agent system architecture.

In the literature, there are already quite a number of architectures for MASs which have been proposed (for example, see [RG92], or [BDJT95]; [WJ95] provides an overview). In our opinion, temporal logic can be used for all of them, to provide a formal semantics and / or to formalize verification and validation proofs. For a number of them, temporal semantics have indeed been defined. In order to make the discussion concrete, we will take one specification language for multi-agent systems, DESIRE (see Subsection 6.2.1), as a case study.

In order to illustrate the formalization and constructions in this section, we will briefly describe an example of a multi-agent system. This example multi-agent model is composed of two co-operative information gathering agents, A and B, and a component EW representing the external world. This system is visualized in Figure 8.1.

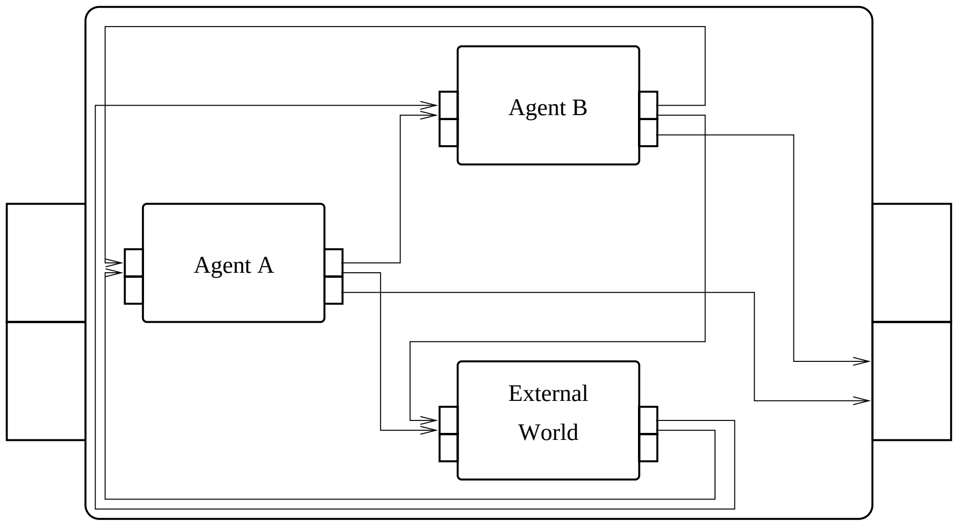

Figure 8.1: Multi-agent system for co-operative information gathering.

Each of the agents is able to acquire partial information about the external world (by observation). Each agent's own observations are insufficient to draw conclusions of a desired type, but the combined information of both agents is sufficient. Therefore, communication is required to be able to draw conclusions. The agents can communicate their own observation results and requests for observation information of the other agent. This (quite common) situation is simplified to the following materialized form. The world situation consists of an object whose shape has to be determined. One agent can only observe the bottom view of the object (e.g., a circle), the other agent the side view (e.g., a square). By exchanging and combining observation information they are able to classify the object (e.g., a cylinder).

Communication from the agent A to B takes place in the following manner:

- Agent A generates at its output interface a statement of the form:

$$\mathsf{to\_be\_communicated\_to}(\langle \mathit{type}\rangle,\ \langle \mathit{atom}\rangle,\ \langle \mathit{sign}\rangle,\ \mathsf{B}).$$

- The information is transferred to B; thereby it translated into

$$\mathsf{communicated\_by}(\langle \mathit{type}\rangle,\ \langle \mathit{atom}\rangle,\ \langle \mathit{sign}\rangle,\ \mathsf{A}).$$

In the example, ⟨*type*⟩ can be filled with a label request (for requesting information from B) or world_info (to provide B with A's information about the world), ⟨*atom*⟩ is an atom expressing information on the world, and ⟨*sign*⟩, is one of pos or neg, to indicate truth or falsity. Interaction between agent A and the world takes place as follows:

- Agent A generates at its output interface a statement of the form:

  to_be_observed(⟨*atom*⟩).

- The information is transferred to EW; thereby it is translated into

  to_be_observed_by(⟨*atom*⟩, A).

- The external world EW generates at its output interface a statement of the form:

  observation_result_for(⟨*atom*⟩, ⟨*sign*⟩, A).

- The information is transferred to A; thereby it is translated into

  observation_result(⟨*atom*⟩, ⟨*sign*⟩).

Part of the output of an agent are conclusions about the classification of the object of the form object_type(s), where s is a shape; these are transferred to the output of the system.

To be able to perform its tasks, each agent is composed of four components, see Figure 8.2: three for generic agent tasks (world_interaction_management, which reasons about the interaction with the outside world, agent_interaction_management, which reasons about the interaction with other agents, and own_process_control, which reasons about the control of the agent itself; in this example it determines the agent characteristics, for example whether the agent is pro-active or reactive), and one for an agent specific task (object_classification). Additionally, there are a number of information links, which have been left out of the picture.

The example used to illustrate the formalization in the current section is restricted to a pro-active agent A and a reactive agent B.

8.1.1 Formalization in temporal logic

Our goal in this subsection is to describe a method to translate a multi-agent system (in DESIRE) into a theory of temporal logic describing its reasoning behavior. So let us first assume that we have a (DESIRE) multi-agent system given. The language in which information can be stated in components, is an order-sorted first-order predicate language. However, the components can only have factual information (that is, their knowledge only contains closed atomic statements: in a component,

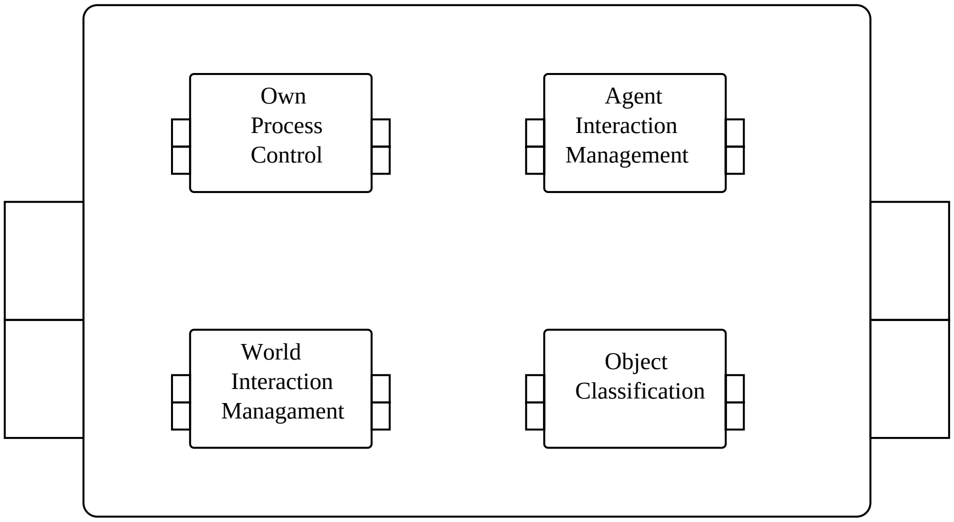

Figure 8.2: Components within the agents.

a fact can be true, false, or unknown). Therefore, we can assume a propositional language. Each component C has a language associated with its input interface (we will denote this language by $\mathcal{L}_{\mathsf{C}}^{in}$), one for its output interface (denoted by $\mathcal{L}_{\mathsf{C}}^{out}$), and it has an internal language (denoted by $\mathcal{L}_{\mathsf{C}}^{int}$). The total language of the component, $\mathcal{L}_{\mathsf{C}}$, is then defined by

$$\mathcal{L}_{\mathsf{C}} = \mathcal{L}_{\mathsf{C}}^{in} \cup \mathcal{L}_{\mathsf{C}}^{int} \cup \mathcal{L}_{\mathsf{C}}^{out}$$

where the three languages are assumed to be disjoint. As the information a component has, is three-valued, we use the information states of IS^{3val} to formalize the state of the input and output interfaces, and the internal state of the agent. We will use temporal partial logic (TPL) to describe the behavior of the components over time. The underlying propositional language should at least contain $\mathcal{L}_{\mathsf{C}}$ for all components C in the system. Additionally, it may contain a language $\mathcal{L}_{\mathsf{E}}$ in which facts that hold in the world outside the system can be described (for describing input and output of the system).

We will now describe a system in temporal partial logic. This will be done compositionally, meaning that a theory will be defined for each component separately. First, consider a composed component C. Then C in a sense 'controls' its internal state (the truth-values of the atoms in $\mathcal{L}_{\mathsf{C}}^{int}$), and its output interface. It also controls the input interfaces of its subcomponents: we make the assumption that when C activates one of its information links, it is C who reads the source interface, and writes onto the target interface. It may read (but does not control) its own input interface and the output interfaces of its subcomponents. Temporal formulae describing C should use the atoms in the language of C, and atoms in the input and output languages of its subcomponents. This motivates the following definition.

Definition 8.1 (Language composition) Let COMPS be a set of component names with a subcomponent relation **sub**, such that COMPS together with **sub** forms a finite tree. For each component $\mathsf{D} \in$ COMPS, define its *interface language*, denoted by $\mathcal{L}^{if}_{\mathsf{D}}$, as

$$\mathcal{L}^{if}_{\mathsf{D}} = \mathcal{L}^{in}_{\mathsf{D}} \cup \mathcal{L}^{out}_{\mathsf{D}},$$

and define its *bridge language*, denoted by $\mathcal{L}^{+}_{\mathsf{D}}$, as

$$\mathcal{L}^{+}_{\mathsf{D}} = \mathcal{L}_{\mathsf{D}} \cup \bigcup_{\mathsf{C}\ \mathbf{sub}\ \mathsf{D}} \mathcal{L}^{if}_{\mathsf{C}}.$$

The collection $(\mathcal{L}^{*}_{\mathsf{D}})_{\mathsf{D} \in \mathrm{COMPS}}$ is inductively defined as

$$\begin{array}{ll} \mathcal{L}^{*}_{\mathsf{D}} = \mathcal{L}_{\mathsf{D}} & \text{if } \mathsf{D} \text{ is primitive} \\ \mathcal{L}^{*}_{\mathsf{D}} = \mathcal{L}_{\mathsf{D}} \cup \bigcup_{\mathsf{C}\ \mathbf{sub}\ \mathsf{D}} \mathcal{L}^{*}_{\mathsf{C}} & \text{if } \mathsf{D} \text{ is composed.} \end{array}$$

Using these bridge languages, we can formulate a temporal theory for each of the components. Let us first focus on a primitive component.

Primitive components: knowledge base

We first have to mention some additional facts. The input, output, and internal state of an agent are in fact divided into two parts: an object level part and a meta-level part. In the object level, the information about which the component is reasoning is contained. This is the language $\mathcal{L}_{\mathsf{C}}$. The meta-level of a component holds information *about* the reasoning process: whether the component is active or not, what its goals are, and also epistemic information (for example, which atoms have been derived to be true or false). A second point is that the internal language contains the input and output languages. We shall denote an atom p in the input language by *input.p*, and the corresponding atom in the internal language by *int.p*, (similarly for the output: there we may have *int.p* and *output.p* for the internal atom and its corresponding output interface atom).

The first thing a primitive component does when it is activated, is to copy its input to its internal state. We assume that the input interface at the meta-level of the component has an atom active that is set to activate the component. The formula describing this action is as follows

$$\begin{array}{l} C(\mathsf{active}) \wedge Y(\neg C(\mathsf{active})) \wedge C(input.l) \to X(C(int.l)) \\ C(\mathsf{active}) \wedge Y(\neg C(\mathsf{active})) \wedge \neg C(input.l) \to X(\neg C(int.l)) \end{array}$$

for all input literals l. The first two parts of these rules express that the component was just activated. After this copying, it sets a control variable to denote that it will now be performing reasoning using its knowledge base:

$$C(\mathsf{active}) \wedge Y(\neg C(\mathsf{active})) \rightarrow X(C(\mathsf{reason})).$$

The knowledge base of the component consists of rules of the form

$$l_1 \wedge \ldots \wedge l_n \rightarrow l$$

where each of the l_i and l may be a literal. The component performs chaining on this set of rules (the negation is interpreted as strong, classical negation, not as negation as failure as in logic programming), computing the minimal three-valued model of the knowledge base. Each such rule can be translated as

$$C(\mathsf{active}) \wedge C(\mathsf{reason}) \wedge XC(int.l_1) \wedge \ldots \wedge XC(int.l_n) \rightarrow XC(int.l).$$

This temporal description makes sure that if reason is true, then the next state is closed with respect to the rules. We have to reset the control variables, and empty the output.

$$\begin{array}{l} C(\mathsf{active}) \wedge C(\mathsf{reason}) \rightarrow \neg X(C(\mathsf{reason})) \wedge X(C(\mathsf{copyout})) \\ C(\mathsf{active}) \wedge C(\mathsf{reason}) \rightarrow \neg X(C(output.l)). \end{array}$$

In the last step, we copy the internal state to the output, and set an output meta-level variable to denote that we are ready.

$$\begin{array}{l} C(\mathsf{active}) \wedge C(\mathsf{copyout}) \wedge C(int.l) \rightarrow X(C(output.l)) \\ C(\mathsf{active}) \wedge C(\mathsf{copyout}) \rightarrow X(\neg C(\mathsf{copyout})) \wedge X(C(\mathsf{ready})). \end{array}$$

This ready fact has to be reset at the beginning, when the component just starts:

$$C(\mathsf{active}) \wedge Y(\neg C(\mathsf{active})) \rightarrow X(\neg C(\mathsf{ready})).$$

In DESIRE, components have a further meta-level predicate, succeeded, the value of which depends on the output literals (on the object level) and on its target. The component has a number of output literals as targets, and its goal may be to derive either as much as possible (indicated by the input literal all_p), or to derive at least one (any), or to derive every literal in the target set, or to derive a literal that was not derived before. So a few of the rules setting succeeded are

$$\begin{array}{l} C(\mathsf{active}) \wedge C(\mathsf{copyout}) \wedge C(\mathsf{all\_p}) \rightarrow X(C(\mathsf{succeeded})) \\ C(\mathsf{active}) \wedge C(\mathsf{copyout}) \wedge C(\mathsf{any}) \rightarrow \\ \qquad\qquad (X(C(\mathsf{succeeded})) \leftrightarrow \bigvee(C(int.l) \wedge C(input.target.l))). \end{array}$$

In fact, the target set and goal also determine which literals are copied to the output, so the rules for copying actually have to be adapted somewhat. There are some other subtle points in DESIRE that have to do with target sets and goals, but this is not the place to go into these issues further.

The approach taken above lets the component compute the total closure of the input facts under the rules, in one time step. It is of course also possible to let the application of a rule take a step in time (similarly to what is done in Section 5.4 on proof systems), and to translate a rule $l_1 \wedge \ldots \wedge l_n \rightarrow l$ into the rule

$$\varphi \wedge C(int.l_1) \wedge \ldots \wedge C(int.l_n) \rightarrow X(C(int.l))$$

where the φ represents the control information. This also allows more finetuning of the process: when the goal of the component is to derive *one* literal of the target set (any is true), then the derivation process can be stopped as soon as a literal is derived. It is even possible to change the formulae so that every application of a single rule takes a single step in time (no two rules fire simultaneously).

To give an example of a rule, consider the primitive component object_classification within agent A. Its knowledge base contains a number of rules to determine the type of the object, based on its own and B's view of the object. One of them is

if view(A, circle) **and** view(B, square) **then** object_type(cylinder).

Its formalization is

$$C(\mathsf{OC.active}) \wedge C(\mathsf{OC.reason}) \wedge X(C(\mathsf{OC}.int.\mathsf{view(A,\ circle)}))\wedge \\ X(C(\mathsf{OC}.int.\mathsf{view(B,\ square)})) \rightarrow X(C(\mathsf{OC}.int.\mathsf{object\_type(cylinder)}))\text{.}$$

We have prefixed the atoms with "OC." to make them unique for the signatures of OC.

Composed component: links and task control

Composed components are composed of other components and links between them. As mentioned earlier, these links transfer facts from an interface of one component to an interface of (usually) another component. The task control knowledge of a composed component may derive that a certain link should be up to date, meaning that the transfer of facts should have been made. We give an example, building on our earlier description of the communication from agent A to agent B. This communication is done via a link communication_from_A_to_B. The task control of the component for the entire system has an internal atom, link_state(communication_from_A_to_B, uptodate). If this atom is derived, the link communication_from_A_to_B should be up to date. This link is described by the following rules setting the facts communicated_by to true

$$Y(C(\mathsf{to\_be\_communicated\_to}(\langle \mathit{type}\rangle,\ \langle \mathit{atom}\rangle,\ \langle \mathit{sign}\rangle,\ \mathsf{B})))\wedge \\ C(\mathsf{link\_state(communication\_from\_A\_to\_B,\ uptodate)}) \rightarrow \\ C(\mathsf{communicated\_by}(\langle \mathit{type}\rangle,\ \langle \mathit{atom}\rangle,\ \langle \mathit{sign}\rangle,\ \mathsf{A})),$$

and rules setting the fact to unknown again:

$$Y(\neg C(\mathsf{to\_be\_communicated\_to}(\langle \mathit{type}\rangle,\ \langle \mathit{atom}\rangle,\ \langle \mathit{sign}\rangle,\ \mathsf{B})))\wedge \\ C(\mathsf{link\_state(communication\_from\_A\_to\_B,\ uptodate)}) \rightarrow \\ \neg C(\mathsf{communicated\_by}(\langle \mathit{type}\rangle,\ \langle \mathit{atom}\rangle,\ \langle \mathit{sign}\rangle,\ \mathsf{A})).$$

If to_be_communicated_to can also be set to false, then the two above types of rules should be copied for the literal ¬to_be_communicated_to. It is assumed here that to_be_communicated_to occurs (only) in the output interface of the component of agent A, and that communicated_by occurs (only) in the input interface of the component of agent B. If an atom occurs in more than one place, it should be replaced by distinct copies.

The task control knowledge consists of rules linking the current and previous state (of the subcomponents) to the next state: some component(s) should be active, and some link(s) should be up to date. Our example multi-agent system may not be too instructive regarding the task control knowledge, as basically all components are active all the time, awaiting new input. But a basic kind of task control rule would be of the form "if component X is just ready, then component Y should be started, and the information link k should be up to date". This can be formalized as

$$C(\mathsf{X}.\mathit{output}.\mathsf{ready}) \wedge Y(\neg C(\mathsf{X}.\mathit{output}.\mathsf{ready}))$$
$$\rightarrow X(C(\mathsf{Y}.\mathit{input}.\mathsf{active})) \wedge X(C(\mathsf{link\_state(communication\_from\_A\_to\_B,\ uptodate)})).$$

Compositionality

Above, a way was sketched to translate parts of a multi-agent system into temporal partial logic (many details were left untreated that would have to be clarified in order to get a complete and faithful translation). This should lead to a temporal theory for each component. If this component is primitive, it consists of the temporal description of its knowledge base. Otherwise, it consists of the temporal description of its task control knowledge, together with the temporal description of all of its links. This leads to a family of temporal theories.

Definition 8.2 Let $(\mathcal{L}^+_{\mathsf{D}})_{\mathsf{D}\in\mathrm{COMPS}}$ be the collection of bridge languages. A *compositional temporal theory* for this collection is a collection $(T_{\mathsf{D}})_{\mathsf{D}\in\mathrm{COMPS}}$ of temporal theories such that each T_{D} is in the language $\mathcal{L}^+_{\mathsf{D}}$. The *collection of cumulative theories* $(T^*_{\mathsf{D}})_{\mathsf{D}\in\mathrm{COMPS}}$ is defined by

$$\begin{array}{ll} T^*_{\mathsf{D}} = T_{\mathsf{D}} \cup \bigcup_{\mathsf{C}\ \mathbf{sub}\ \mathsf{D}} T^*_{\mathsf{C}} & \text{if D is a composed component} \\ T^*_{\mathsf{D}} = T_{\mathsf{D}} & \text{if D is a primitive component.} \end{array}$$

The theory T^*_{D} should provide a complete description of what T^*_{D}'s behavior is (with its subcomponents).

8.1.2 Persistence

The reader may have noticed that in the temporal descriptions above, only the changes of information were described. There were no formulae indicating that some things should remain the same. But in DESIRE, as is the case in many systems, information does not change at random: there is a *default persistence* of information.

If we just consider the temporal partial models of the temporal description, however, information is allowed to change without a cause for its change. Of course, we do not want to leave the burden of describing when information does *not* change to the person who makes the temporal specification. There are at least two possible solutions to this problem.

First of all, we have in fact seen a number of situations where information should change as little as possible, only when there is a rule stating it should change. This was the case in the temporal descriptions of, for instance, default logic. So, one solution is to consider only those models of a description where information change is minimal. For this purpose, we can use the preferential ordering $\preceq^{sc}$ of sequential minimal change, defined in Section 4.4.3, and the associated minimal consequence relation $\models_{\preceq^{sc}}$ to infer properties of the system. This provides a minimal change semantics to any description of a system.

Another solution is to provide a method by which a temporal specification of a system (describing only the changes) can be transformed (mechanically) into a description that takes into account the minimal change. The temporal partial models of the new description are the intended models of the original theory. An example of this kind of semantics is the *completion* semantics of logic programs. The advantage of this solution is that ordinary, classical theorem provers can be used on the resulting theory (whereas the first solution requires tools especially made for $\models_{\preceq^{sc}}$). In this section, we will describe such a semantics.

The *temporal completion* method we will describe, works only for a restricted fragment of temporal partial logic. Fortunately, the rules described so far fall within this fragment. It consists of a kind of executable temporal formulae. Roughly spoken, executable temporal formulae are temporal formulae of the form

declarative past implies *imperative future*

For more details on this paradigm, and the different variants within, see, apart from Section 6.1 in this thesis, also [BFGH91, BFG+96]. We will consider here an even simpler form, called *simplified executable temporal formulae*:

past and present $\to$ present

where 'past' and 'present' are further restricted, see below. Two remarks are in order here. First of all, the rules described so far are not in this format, they are of the format 'previous and current and next imply next'. But a simple shift in the current time point transforms this into 'two time points ago and previous and current imply current'. We feel the completion is slightly easier to understand for rules in the format of simplified executable temporal formulae. Secondly, we allow rules with both precondition and conclusion referring to the present. This is to allow changes to happen instantaneously. A rule like $C(a) \to C(b)$ expresses the constraint that if a is set to true, then b should be true as well. The formalization of a knowledge base refers to facts at the same point in time in the precondition and conclusion of a rule (referring to the next point in time, but see the first remark). A formal definition of simplified executable temporal formulae follows.

Definition 8.3 (Simplified executable temporal formulae) A *simplified executable temporal formula* is a temporal formula F of the form

$$\varphi \to \psi$$

where the right hand side, ψ, is called the *head*, denoted by head(F), and is taken from the set *HEADS*, defined as

$$\begin{array}{rcl} \mathit{HEADS} & = & \{C(l) \mid l \text{ literal}\} \cup \\ & & \{\neg Ca \wedge \neg C\neg a \mid a \text{ atom}\}. \end{array}$$

The left hand side, φ is called the *body*, and is denoted by body(F). It must be a conjunction of temporal literals of the following form: $YYCl$, $\neg YYCl$, YCl, $\neg YCl$, Cl, $\neg Cl$, where l is a propositional literal.

We can define a form of Clark's completion on simplified executable temporal formulae.

Definition 8.4 (Temporal completion) Let $\mathfrak{T}$ be a temporal theory consisting of simplified executable temporal formulae. For each $h \in \mathit{HEADS}$, let

$$\mathfrak{T}^h = \{F \in \mathfrak{T} \mid \text{head}(F) = h\}.$$

For a literal l, define

$$\mathsf{tc}(\mathfrak{T}^{Cl}) = \left[\begin{array}{l} \bigvee\{\text{body}(F) \mid F \in \mathfrak{T}^{Cl}\} \vee \\ \left(\begin{array}{l} \neg \bigvee\{\text{body}(F) \mid F \in \mathfrak{T}^{C\neg l}\} \wedge \\ \neg \bigvee\{\text{body}(F) \mid F \in \mathfrak{T}^{\neg Cl \wedge \neg C\neg l}\} \wedge \\ YCl \end{array} \right) \end{array} \right] \leftrightarrow \quad Cl.$$

For an atom a, define

$$\mathsf{tc}(\mathfrak{T}^{\neg Ca \wedge \neg C\neg a}) = \left[\begin{array}{l} \bigvee\{\text{body}(F) \mid F \in \mathfrak{T}^{\neg Ca \wedge \neg C\neg a}\} \vee \\ \left(\begin{array}{l} \neg \bigvee\{\text{body}(F) \mid F \in \mathfrak{T}^{Ca}\} \wedge \\ \neg \bigvee\{\text{body}(F) \mid F \in \mathfrak{T}^{C\neg a}\} \wedge \\ \neg YCa \wedge \neg YC\neg a \end{array} \right) \end{array} \right] \leftrightarrow \quad \neg Ca \wedge \neg C\neg a.$$

The intuition behind this definition is that an atom a should have a certain truth value (expressed as Ca, $C\neg a$ or $\neg Ca \wedge \neg C\neg a$), just in case there is either an applicable rule setting it to this value, or it had this value in the previous moment, and there is no rule applicable that could change it.

It may seem that a rule for $\neg Ca \wedge \neg C\neg a$ is unnecessary, as the rules for Ca and $C\neg a$ already determine what $\neg Ca \wedge \neg C\neg a$ should be. In general this is true, except in the situation that there is a rule for Ca (or for $C\neg a$) which is applicable, as well as an applicable rule with head $\neg Ca \wedge \neg C\neg a$. The extra $\neg Ca \wedge \neg C\neg a$ temporal completion then eliminates such a model. Without this rule, the Ca rule takes precedence.

Definition 8.5 (Temporal completion, continued) Let $\mathfrak{T}$ be a temporal theory consisting of simplified executable temporal formulae. The *temporal completion* of $\mathfrak{T}$ is defined by

$$\begin{array}{rcl}\mathsf{tc}(\mathfrak{T}) & = & \{\mathsf{tc}(\mathfrak{T}^{Cl}) \mid l \text{ literal } \} \cup \\ & & \{\mathsf{tc}(\mathfrak{T}^{\neg Ca \wedge \neg C \neg a}) \mid a \text{ an atom}\}.\end{array}$$

So the temporal partial models of $\mathsf{tc}(\mathfrak{T})$ are the intended, minimal change models of $\mathfrak{T}$. One may wonder what the relation is with the SCTEL semantics, which were defined to model minimal change. The question is: for a theory $\mathfrak{T}$, what is the relation between the $\preceq^{sc}$-minimal models of $\mathfrak{T}$, and the temporal partial models of $\mathsf{tc}(\mathfrak{T})$? In general, they may be different. Consider, for example, the theory $T = \{YCa \wedge Ca \rightarrow Cb, \neg YCa \rightarrow Ca\}$. The completion of T contains the formulae $(YCa \wedge Ca) \vee YCb \leftrightarrow Cb$ and $\neg YCa \vee YCa \leftrightarrow Ca$ (as well as formulae for $C\neg a$ and $C\neg b$). This means that in any model of the completion of T, the atom a is always true, and therefore, the atom b is true from time point 1 onwards (and not in time point 0). This model is also a $\preceq^{sc}$-minimal model of T. However, there are more $\preceq^{sc}$-minimal models of T, in which a is true at time point 0, and nothing is true at time point 1. The rule $YCa \wedge Ca \rightarrow Cb$ may have the effect of setting a to unknown at time point 1, instead of setting b to true, as intended. This behavior only occurs when there are rules with a C-atom occurring in the left hand side of a rule. Some of the formulae used before to describe the behavior of a multi-agent system were of this form (modulo the shift in time). The formulae describing the application of the knowledge base of a primitive component, however, can be changed so that this application is not done in one time step, but takes time, as remarked earlier. The resulting formulae do not contain C-atoms on the left hand side. The same can be done for the formulae describing the application of a link (using extra control variables). We have to assume one extra condition on the theory describing a multi-agent system: whenever the left hand sides of a subset of formulae are jointly satisfiable (in TPL), then their corresponding right hand sides must also be jointly satisfiable. If we have made a correct and complete theory (in the sense that in every temporal situation, the theory describes what the system will do) then this requirement will be satisfied, and our completion semantics is the same as the SCTEL semantics, with one proviso. The SCTEL semantics allow any first state that satisfies the rules, whereas in the completion the first state is determined by the rules satisfied in time point 0. As an example, if we consider the rule $\neg YCa \rightarrow Ca$, then there is just one model of the completion, in which a is true from the beginning, and all other atoms remain unknown. The SCTEL semantics also allow the model in which both a and b are known from the start (this model does not satisfy the formula $YCb \leftrightarrow Cb$ of the completion). This behavior could easily be built into the completion by changing a formula $\mathsf{tc}(\mathfrak{T}^{Cl})$ into $\neg Y\top \rightarrow \mathsf{tc}(\mathfrak{T}^{Cl})$ (making it applicable only at time points greater than zero) and adding T itself to the completion (to ensure that time point zero satisfies T).

Proposition 8.6 Let $\mathfrak{T}$ be a temporal theory of simplified executable temporal formulae, such that for each $F \in \mathfrak{T}$, body(F) does not contain temporal literals of the form Cl or $\neg Cl$. Suppose that for any $S \subseteq \mathfrak{T}$, if $\{\text{body}(F) \mid F \in S\}$ is TPL-satisfiable, then $\{\text{head}(F) \mid F \in S\}$ is TPL-satisfiable. Then for any TPL-model $\mathcal{M}$ the following are equivalent:

1. $\mathcal{M}$ is a model of $\mathsf{tc}(\mathfrak{T})$.
2. $\mathcal{M}$ is a $\preceq^{sc}$-minimal model of $\mathfrak{T}$ and $\mathcal{M}_0$ is a $\preceq$-minimal model of the right hand sides of the rules in $\mathfrak{T}$ applicable at time point zero.

Proof: Use the first and third statements of Proposition 4.61. The completion is the formalization in TPL of the third statement of that proposition. □

We have placed considerable emphasis on the compositional nature a description of a compositional system should have. Each component C has its own theory, T_{C}. Taking the temporal completion, however, can destroy the compositionality. Let us take an example component hierarchy consisting of three components, A, B, and C with C **sub** B **sub** A. Suppose that $p \in \mathcal{L}_{\mathsf{A}}, q \in \mathcal{L}_{\mathsf{B}}^{if}$, and $r \in \mathcal{L}_{\mathsf{C}}^{if}$. Furthermore, suppose there is one rule, $Y(Cp) \rightarrow Cq$, in T_{A} (note that this rule is in $\mathcal{L}_{\mathsf{A}}^{+}$!) and that there is one rule, $Y(Cr) \rightarrow Cq$, in T_{B} (again, note that this rule is in $\mathcal{L}_{\mathsf{B}}^{+}$). When taking the temporal completion of T_{A} and T_{B} separately, we end up with

$$Y(Cp) \vee Y(Cq) \leftrightarrow Cq$$

as well as

$$Y(Cr) \vee Y(Cq) \leftrightarrow Cq$$

which may obviously lead to undesired results. On the other hand, if we take the completion of the union of all theories, T_{A}^{*}, we get a rule

$$Y(Cp) \vee Y(Cr) \vee Y(Cq) \leftrightarrow Cq$$

which is the intended formula defining Cq, but which is neither in $\mathcal{L}_{\mathsf{A}}^{+}$, nor in $\mathcal{L}_{\mathsf{B}}^{+}$, destroying compositionality. The example is constructed in such a way that an atom (of B) directly depends on information in a higher component (A), as well as on information in a lower component (C). By inspecting the temporal description of components in DESIRE, it can be seen that this situation does not occur: the input atoms of a component are set ('controlled') by its super-component, whereas it sets ('controls') its own internal and output atoms. Therefore, for every atom a, the rules defining it ($\mathfrak{T}^{Ca}$, $\mathfrak{T}^{C\neg a}$ and $\mathfrak{T}^{\neg Ca \wedge \neg C \neg a}$) are all contained in the theory of *one* component. Under the assumption that all rules defining an atom are contained in T_{C} for some component C, the problem above does not occur, as the temporal completion of the union of all theories is the same as the union of the completions of the individual theories:

$$\mathsf{tc}(T_{\mathsf{A}}^{*}) = \bigcup_{\mathsf{C} \in \text{COMPS}} \mathsf{tc}(T_{\mathsf{C}}).$$

One of the goals of translating DESIRE specifications into temporal logic is to provide a formal (temporal) semantics to the former (see [BTWW96]). The other reason is that it allows the formalization of proofs of properties of a system (for verification and validation purposes). The next subsection is devoted to the second subject.

8.1.3 Compositional verification

Compositional theories describing a compositional system allow compositional verification. In [CJT97, JT97] a compositional verification method is described and applied to diagnostic reasoning and multi-agent systems, respectively; we will briefly summarize this method. Remember that we have a hierarchy described by the set of components, COMPS, and a subcomponent relation **sub**. This hierarchy should form a (finite) tree. Each component in this hierarchy is part of a *level* (also called *level of abstraction*). The top component is at level 0 (denoted L_0), its descendants are of level 1 (L_1), and so on. In general, if the distance from a component to the top component is n, then the component is of level n.

A. Verifying one abstraction level against the other
For each abstraction level the following procedure is followed:

1. Determine which properties are of interest for the (higher level) component D.
2. Determine assumptions (properties for the lower level components) that guarantee D's properties.
3. Prove D's properties on the basis of the properties of its subcomponents.

B. Verifying a primitive component
For primitive knowledge-based components a number of verification techniques exist in the literature, see for example [TW94].

C. The overall verification process
To verify the complete system:

1. Determine the properties are that are desired for the whole system.
2. Apply the above procedure **A** iteratively.
   In the iteration the desired properties of abstraction level L_i are either:
   - those determined in step **A**(1), if $i = 0$, or
   - the assumptions made for the higher level L_{i-1}, if $i > 0$.
3. Verify the primitive components according to **B**.

The results of verification are:

- Properties and assumptions at the different abstraction levels.

- Logical relations between the properties of different abstraction levels.

Both static and dynamic properties and connections between them are covered. Furthermore, process and information hiding limits the complexity of the verification per abstraction level (see [JT97]). An example of some properties of different components for our multi-agent system is given in Figure 8.3.

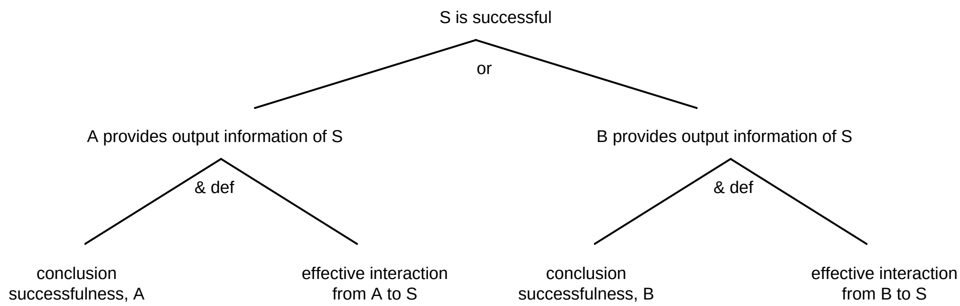

Figure 8.3: Logical relations between properties of components.

One can also distinguish dependencies between properties within one level, see Figure 8.4.

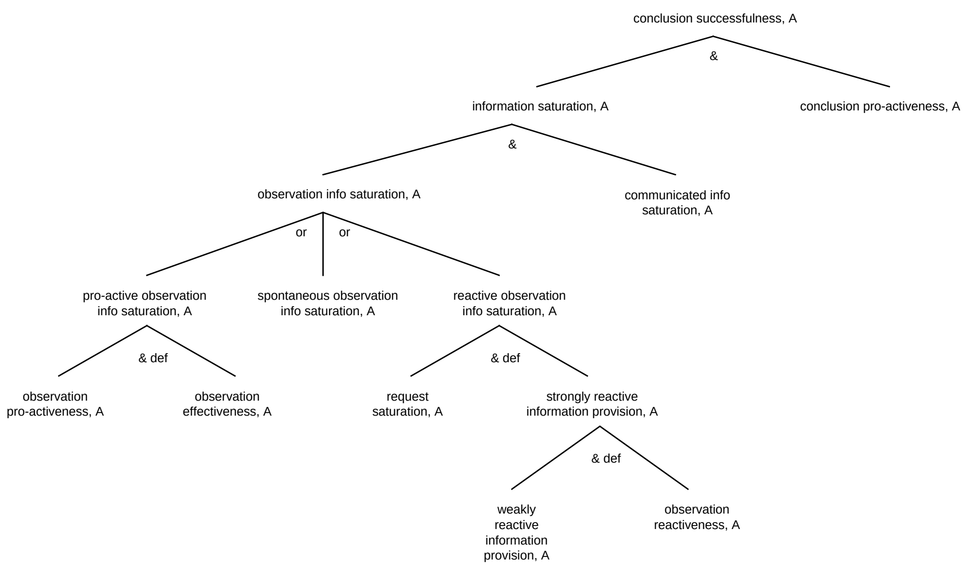

Figure 8.4: Logical relations between properties of one component.

These pictures (from [JT97]) mention properties (of components) which have a formal (mathematical) definition. If we want to use temporal logic to formalize the verification process, these properties should be expressible in temporal logic, and this indeed turns out to be the case. To give just one example, the property of 'observation pro-activeness of A' (see Figure 8.4) means that agent A will eventually try to observe the object whose type should be determined, independent of its input. This can be formalized by the following formula

$$\bigwedge_r F(C(\mathsf{to\_be\_observed}(\mathsf{view}(A,r))))$$

where r ranges over the possible shapes. This means that A wants to know, for every shape, whether that shape is observed or not (usually the answer will be that just one is observed, and the others are not). The other properties can be formalized as well (see [JT97] for a formalization in 'general' mathematical notation; the translation into temporal logic is rather straightforward). The compositional verification process needs compositional proofs.

Definition 8.7 A *composition of properties* for the collection of languages $(\mathcal{L}_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$ is a collection $(P_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$ where for each C the set P_C is a set of temporal formulae in the language $\mathcal{L}^{if}_\mathsf{C}$.

Given the theories describing the components, and the properties of components, we can define compositional provability.

Definition 8.8 (Compositional and global provability) For a collection of languages $(\mathcal{L}_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$, let a composition of properties $(P_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$ and a compositional temporal theory $(T_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$ be given. Let $\mathrel{|\!\!\sim}_{tpl}$ be an entailment relation for temporal partial logic.

1. The composition of properties $(P_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$ is *compositionally provable* with respect to $\mathrel{|\!\!\sim}_{tpl}$ from the compositional temporal theory $(T_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$, if for each component D the following holds:

$$\begin{array}{ll} T_\mathsf{D} \cup \bigcup_{\mathsf{C}\ \mathbf{sub}\ \mathsf{D}} P_\mathsf{C} \mathrel{|\!\!\sim}_{tpl} P_\mathsf{D} & \text{if D is composed} \\ T_\mathsf{D} \mathrel{|\!\!\sim}_{tpl} P_\mathsf{D} & \text{if D is primitive.} \end{array}$$

2. The composition of properties $(P_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$ is *globally provable* with respect to $\mathrel{|\!\!\sim}_{tpl}$ from the compositional temporal theory $(T_\mathsf{C})_{\mathsf{C}\in\mathrm{COMPS}}$, if for each component D the following holds:

$$T^*_\mathsf{D} \mathrel{|\!\!\sim}_{tpl} P_\mathsf{D}.$$

Compositional provability does not necessarily imply global provability. However, the implication holds if the entailment relation satisfies, apart from reflexivity (if $V \subseteq W$, then $W \mathrel{|\!\!\sim}_{tpl} V$), the property of transitivity:

$$T \mathrel{|\!\!\sim}_{tpl} U \;\&\; U \mathrel{|\!\!\sim}_{tpl} W \;\Rightarrow\; T \mathrel{|\!\!\sim}_{tpl} W \qquad (\textit{Transitivity})$$

for all sets of formulae T, U, W. It is well-known that transitivity and reflexivity imply monotonicity.

Proposition 8.9 If the entailment relation $\mathrel{|\!\!\sim}_{tpl}$ satisfies reflexivity and transitivity then compositional provability with respect to $\mathrel{|\!\!\sim}_{tpl}$ implies global provability with respect to $\mathrel{|\!\!\sim}_{tpl}$.

So in particular any sound classical proof system for TPL satisfies this requirement (see Section 9.1.1 for a classical proof system sound and complete for TEL). The complete procedure we envisage is the following. Suppose we have a specification (for example in DESIRE) of a compositional multi-agent system. This specification can be (automatically) translated into a compositional temporal theory $(T_{\mathsf{C}})_{\mathsf{C} \in \mathrm{COMPS}}$. Taking the temporal completions (also automatically) leads to $(\mathsf{tc}(T_{\mathsf{C}}))_{\mathsf{C} \in \mathrm{COMPS}}$. Then the compositional verification method is applied to this compositional theory, using a classical proof system which is sound (and preferably complete) for TPL. Tools can (partly) automate the process of finding and / or verifying the proofs.

8.1.4 Conclusions and related work

The compositional verification method formalized in this section can be applied to a broad class of multi-agent systems. Compositional verification for one abstraction level deep is based on the following very general assumptions:

1. A multi-agent system consists of a number of agents and external world components.
2. Agents and components have explicitly defined input and output interface languages; all other information is hidden; information exchange between components can only take place via the interfaces (information hiding).
3. A formal description exists of the manner in which agents and world components are composed to form the whole multi-agent system (composition relation).
4. The semantics of the system can be described by the evolution of states of the agents and components at the different levels of abstraction (state-based semantics).

This non-iterative form of compositional verification can be applied to many existing approaches, for example, to systems designed using Concurrent METATEM [Fis94, FW97]. Compositional verification involving more abstraction levels assumes, in addition:

1. Each agent or component is composed of subcomponents or it is described by a knowledge base.

2. A formal description exists of the manner in which agents or components are composed of subcomponents (composition relation).

3. Information exchange between components is only possible between two components at the same or adjacent levels (information hiding).

Currently not many approaches to multi-agent system design exist that exploit iterative compositionality. One approach that does is the compositional development method for multi-agent systems DESIRE. The compositional verification method formalized in this section fits well to DESIRE, but not exclusively.

Two main advantages of a compositional approach to modeling are the transparent structure of the design and support for reuse of components and generic models. The compositional verification method extends these main advantages to (1) a well-structured verification process, and (2) the reusability of proofs for properties of components that are reused.

The first advantage entails that both conceptually and computationally the complexity of the verification process can be handled by compositionality at different levels of abstraction. Apart from the work reported in [JT97], a generic model for diagnosis has been verified [CJT97] and a multi-agent system with agents negotiating about load-balancing of electricity use [BCG$^+$98]. The second advantage entails: if a modified component satisfies the same properties as the previous one, the proof of the properties at the higher levels of abstraction can be reused to show that the new system has the same properties as the original. This has high value for a library of reusable generic models and components. The verification of generic models forces one to find the assumptions under which the generic model is applicable for the considered domain, as is also discussed in [FSGW96]. A library of reusable components and generic models may consist of both specifications of the components and models, and their design rationale. As part of the design rationale, at least the properties of the components and their logical relations can be documented.

The usefulness of temporal partial logic was investigated to formalize verification proofs. As a test, the properties and proofs that were found for verification of an example multi-agent system for co-operative information gathering [JT97] were successfully formalized within this logic. Our study shows that temporal partial logic provides enough expressivity for dynamics and reasoning about time, and formalizes incomplete information states in an adequate manner. To obtain the right structure in accordance with the compositional system design, the logic is equipped with a number of compositional structures: compositions of sublanguages, compositional theories, and compositional provability. It was established that under the assumption that the provability relation is reflexive and transitive, compositional provability implies global provability. Therefore this logic is adequate if the executable temporal theories formalizing a specification are temporally completed, a temporal variant of Clark's completion [Cla78] for logic programs. In this case classical provability can be used, which is more transparent than the more complicated non-classical provability relations that are possible. In this section, the components are all supposed to have a different language. When an atom needs to be used in two different components, then two copies have to be made, one for each component. A more structured way is to introduce operators for the different components (or agents): instead of the single C operator, we could define different C operators for different components. Within one component, we may have that an atom is known in the input, internally, or in the output. Therefore, in a recent paper ([EJT98]) we introduced operators Cin_X, $Cint_X$, and $Cout_X$ for the input, internal and output information known to component X. On the semantics side, this is paralleled by structured information states.

Let us briefly come back to the temporal belief logic (TBL) of [FW97] that was used to define semantics and verify properties for systems specified in Concurrent METATEM [Fis94]. A difference with our temporal logics mentioned earlier lies in the fact that TBL uses modal operators to distinguish knowledge of different agents; as mentioned, our logic can easily be transformed to do the same [EJT98]. Other than that, TBL and temporal partial logic both use the natural numbers as flow of time. A main difference in comparison to [FW97] is that our approach exploits compositionality. In Concurrent METATEM no iterated compositional structures can be defined, as is the case in DESIRE. Therefore verification in TBL always takes place at the global level, instead of the iterated compositional approach to verification described in this section. Another difference is that in our approach the states in the base logic are in principle three-valued, whereas the states in Concurrent METATEM are two-valued: an atom in a state that is not true is assumed false in this state.

We already mentioned the use of temporal logic for specification and verification of processes in theoretical computer science. The compositional approach to verification has been recognized as valuable in that area as well (see for example [Cha95]).

A future continuation of this work will consider the development of tools for compositional verification. To support the hand-work of verification it would be useful to have tools to assist in the creation of the proof.

8.2 Approximate classification

In this section, we will apply some of the theory developed in previous chapters to formalize the analysis of approximate classification. The multiple belief state operators defined in Chapter 2 capture the phenomenon that the reasoning of an agent, starting from an initial set of beliefs, may lead to a number of different information states, each holding a possible set of conclusions. In the examples, this multiplicity of outcomes was usually attributable to incompleteness, vagueness or uncertainty of the input. The set of initial beliefs is enlarged, or (in the case of belief revisions) is reduced. So the set of inputs is altered, and this can often be done in more than one way. In this section, we will add a second aspect to this, namely interpretation: information coming from the outside world (observations) are to be given a meaning. In logic the notion of interpretation mapping has been introduced to describe the interpretation of one logical theory in another logical theory, for example geometry in algebra (cf. Chapter 5 in [Hod93]). This notion assumes a choice for one interpretation, and does not cover cases in which multiple interpretations at the same time are relevant. Therefore, we will introduce multi-interpretation operators and apply them to formalize multiple interpretations of observation information. The notion of multi-interpretation operator is rather general: it subsumes on the one hand the notion of interpretation in logic, and on the other hand the notion of a multiple belief state operator (using the information state frame IS^{syn}). The main difference between multi-interpretation operators and multiple belief state operator based on IS^{syn}, is that the input language of a multi-interpretation operator may be different from the language in which the possible belief sets (a better word would be 'interpretations') are expressed.

A specific type of multi-interpretation operator is defined to interpret observation information in approximate classification tasks. The generic task formalized by such an operator is as follows. Suppose there is an object in the world, and one is interested in the values of attributes of this object. It is possible to observe the object leading to input information consisting of observable properties. On the basis of these properties information on the values of attributes of the object is derived. This task involves interpretation: interpreting observable properties in terms of values of attributes (which may be difficult or impossible to observe directly).

Two problems occurring often in such classification tasks in real-world domains are underspecification and overspecification. Underspecification occurs when the observations are sufficient to exclude some of the values of attributes, but insufficient to determine unique values for each of the attributes: a range of values may still be possible. Overspecification occurs when the observation information is contradictory: for some of the attributes no value is possible. Underspecification can lead to an approximation (an upper bound) of the solution of the classification: a set of possibilities, one of which is the right solution. If the number of observations increases, the approximation may come closer to a unique solution: the resulting sets of possible classifications will decrease with the increase of observation information. Overspecification leads to a trivial approximation from the other direction: the empty set as a lower bound (no classification at all). The combination of underspecification and overspecification as it occurs often in practical domains is problematic. The occurrence of contradictory observation information interferes with the approximations that can be used as upper bound of the solution.

Multi-interpretation operators can be used to clarify this interference: such an operator formalizes the phenomenon that there is more than one possibility of interpreting the observed findings. A generic multi-interpretation operator is introduced to formalize such tasks. The input language of the operator is restricted to observation information only; interpretations of this observation information are expressed in terms of the output language of the operator. This formalization identifies and separates the overspecification and underspecification and entails an approximate solution of a classification problem in the form of multiple approximations.

One domain in which multi-interpretable observations can be analyzed using a technique based on the distinction of different views, is the domain of ecology. Here the possible values of abiotic factors such as moisture and acidity of a terrain, are determined on the basis of the plant species found on the terrain.

We will first introduce multi-interpretation operators and selective interpretation operators and some properties they may have.

8.2.1 Multi-interpretation operators and approximate classification

In this subsection the notion of multi-interpretation operator is introduced, a specific type of multi-interpretation operator is defined that formalizes approximate classification, and some properties of this multi-interpretation operator are proven.

Multi-interpretation operators

A multi-interpretation operator is an operator that assigns to each set of input information a set of interpretations. The input information is described by propositional formulae in a propositional language $\mathcal{L}_1$. An interpretation is a set of propositional formulae, which is closed under the standard propositional consequence operator Cn. Such a closed set will be called a *belief set*, and we assume that they are based on a (possibly different) propositional language $\mathcal{L}_2$. A belief set can be seen as a possible set of beliefs of an agent with perfect (propositional) reasoning capabilities. The definition of multi-interpretation operator, along with some properties, is analogous to the case of multiple belief state operators for IS^{syn}, with the exception of the difference in input and output language.

Definition 8.10 (Multi-interpretation operator)

1. A *multi-interpretation operator* MI with input language $\mathcal{L}_1$ and output language $\mathcal{L}_2$ is a function $MI : \mathcal{P}(\mathcal{L}_1) \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{L}_2))$ that assigns a set of belief sets to each set of input facts.
2. A multi-interpretation operator MI satisfies *non-inclusiveness* if for all $X \subseteq \mathcal{L}_1$ and all $S, T \in MI(X)$, if $S \subseteq T$ then $S = T$.
3. The *kernel* $K_{MI} : \mathcal{P}(\mathcal{L}_1) \rightarrow \mathcal{P}(\mathcal{L}_2)$ of MI is defined by $K_{MI}(X) = \bigcap MI(X)$ for all $X \subseteq \mathcal{L}_1$.
4. If $\mathcal{L}_1 \subseteq \mathcal{L}_2$, then a multi- interpretation operator MI satisfies *inclusion* if for all $X \subseteq \mathcal{L}_1$ and all $T \in MI(X)$ it holds $X \subseteq T$.

If $\mathcal{L}_1 = \mathcal{L}_2$, then it is easy to verify that a multi-interpretation operator is a multiple belief state operator in IS^{syn}. The kernel is then just the final belief state operator associated to the multiple belief state operator. In fact, by taking $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$, a multi-interpretation operator can be seen as a partial multiple belief state operator (defined only on those input sets $X \subseteq \mathcal{L}$ for which $X \subseteq \mathcal{L}_1$). Note that when $MI(X)$ has exactly one element this means that the set $X \subseteq \mathcal{L}_1$ has a unique interpretation under MI.

Selection operators (see Definition 2.16) can be used to select a subset of the possible interpretations in $MI(X)$.

Definition 8.11 (Selective interpretation operator) A *selective interpretation operator for* the multi-interpretation operator MI is a function $C : \mathcal{P}(\mathcal{L}_1) \to \mathcal{P}(\mathcal{L}_2)$ that assigns a belief set to each set of facts, such that for all $X \subseteq \mathcal{L}_1$ it holds that $C(X) \in MI(X)$.

An operator $\mathcal{P}(\mathcal{L}_1) \to \mathcal{P}(\mathcal{L}_2)$ is the analogue of a final belief state operator. It is straightforward to check that if $s : \mathcal{P}(\mathcal{P}(\mathcal{L}_2)) \to \mathcal{P}(\mathcal{P}(\mathcal{L}_2))$ is a single-valued selection operator, then a selective interpretation operator C for a multi-interpretation operator MI can be defined by setting $C(X) = s(MI(X))$ for all $X \subseteq \mathcal{L}_1$.

Below, we will describe a generic type of operator applicable for a specific classification task.

A multi-interpretation operator for approximate classification

Suppose we have an object in the real world (a car, for example), and we are interested in the values of certain attributes of this object (such as the amount of horsepower of the engine). All we can do is observe a number of properties of the object (such as the color, or maybe that it is a Ford). Knowledge relating observable properties to the possible values of attributes is needed to perform this classification task. Using this knowledge, for each attribute certain values can be excluded. In a situation of underspecification for each of the attributes this results in a remaining range of possible values. However, if also overspecification occurs, then in a classical manner it can be derived that for a certain attribute no value at all is possible, which is a contradiction.

A formalization of this approximate classification task can be made using the notions defined above. The language $\mathcal{L}_1$ is the propositional language of which the atoms are the ground atoms defined by the following signature:

- A finite set *Props* of property names: $p_1, \ldots, p_k$.
- A unary predicate: $\mathsf{observed}$.

The meaning of $\mathsf{observed}(p_i)$ is (not surprisingly) that the property p_i has been observed of the object. A variable over the set *Props* will be denoted by P.

The language $\mathcal{L}_2$ is the propositional language extending $\mathcal{L}_1$, of which the additional atoms are the ground atoms defined by the following signature:

- A finite set of attribute names: $a_1, \ldots, a_m$.

- A finite set of values for each of the attributes:

$$\begin{array}{ccc} v_{1,1}, & \dots & ,v_{1,k_1}, \\ & \vdots & \\ v_{m,1}, & \dots & ,v_{m,k_m}. \end{array}$$

A variable over attributes will be denoted by A; a variable over values will be denoted by V.

- Predicates:

$$\begin{array}{l} \mathsf{is\_incompatible\_with}(P, A, V) \\ \mathsf{has\_value}(A, V) \\ \mathsf{is\_indicative}(P). \end{array}$$

The basic idea is that certain (observed) properties may rule out certain values for certain attributes. A fact $\mathsf{is\_incompatible\_with}(P, A, V)$ means that if the observed object has property P, then the attribute A of the object can not have the value V. The predicate $\mathsf{has\_value}(A, V)$ means that attribute A of the object has value V. The last predicate requires a bit more explanation. The basic assumption on the domain is that we may have (potentially) many observations, which can be contradictory. That is, two (or more) observed properties may both rule out values for one attribute, such that together they rule out *all* possible values of that attribute. This may happen for a number of reasons. It may be that our observations are fallible: sometimes we observe a property the object does not have. It is also possible that our knowledge about which properties are incompatible with which values of attributes is uncertain or even not completely correct. Another possibility is that the object is not strictly delineated or strictly homogeneous with respect to its attributes, and some properties are observed from different parts of the object. To deal with this situation, we may label some observed properties as being *indicative*. If the observations are uncertain, 'indicative' may simply mean 'assumed true'. If the object is not homogeneous, then an indicative property is a property related to the view on the object we are interested in. The idea is that some properties are used to infer the values of attributes (in this sense they are 'indicative' of these values), whereas the others are for example wrong, not of interest or coincidental for this view.

There is a knowledge base, KB, in language $\mathcal{L}_2$, that consists of propositional formulae expressing knowledge which is of the following form:

- A (large) number of ground instances of:

  $$\mathsf{is\_incompatible\_with}(P, A, V).$$

  These instances represent the experts' knowledge of which properties rule out which values of certain attributes.

- All ground instances of the generic rule

$$\text{is\_indicative}(P) \wedge \text{is\_incompatible\_with}(P, A, V) \rightarrow \neg\text{has\_value}(A, V).$$

  This rule makes it possible to conclude that certain attributes of the object do not have a certain value. This derivation can be made if an indicative property has been found that does not (generally) occur in objects for which the attribute A has value V.

- Statements expressing that for each attribute at least one value should apply

$$\begin{array}{lcl} \text{has\_value}(a_1, v_{1,1}) \vee & \ldots & \vee\ \text{has\_value}(a_1, v_{1,k_1}) \\ & \vdots & \\ \text{has\_value}(a_m, v_{m,1}) \vee & \ldots & \vee\ \text{has\_value}(a_m, v_{m,k_m}). \end{array}$$

For a given set of observed properties $OBS \subseteq Props$, i.e., input of the form

$$\{\text{observed}(p) \mid p \in OBS\}$$

the set

$$X = KB \cup \{\text{is\_indicative}(p) \mid p \in OBS\}$$

may be inconsistent. That is, it may be inconsistent to assume that all observed properties are indicative for the object. This may occur if there is an attribute A such that for all of its possible values $v_{j,k}$, a property P is observed that negatively indicates this value (which means we have both $\text{is\_indicative}(P)$ and $\text{is\_incompatible\_with}(P, A, v_{j,k})$). With the generic rule, the conclusion $\neg\text{has\_value}(A, v_{j,k})$ is drawn for all possible values $v_{j,k}$ of A. But this is inconsistent with the statement

$$\text{has\_value}(A, v_{j,1}) \vee \ldots \vee \text{has\_value}(A, v_{j,k_j})$$

which is in KB. However, the set of maximal indicative subsets consistent with KB may be considered. This is defined as follows:

Definition 8.12 (Maximal indicative subset)

1. A set of properties $S \subseteq Props$ is an *indicative set of properties* if the theory

$$KB \cup \{\text{is\_indicative}(p) \mid p \in S\}$$

   is consistent.

2. Let $OBS \subseteq Props$ be a given set of observed properties. A set $S \subseteq OBS$ is a *maximal indicative subset of OBS* if it is an indicative set of properties and for each indicative set of properties T with $S \subseteq T \subseteq OBS$ it holds $S = T$. The *set of maximal indicative subsets* of OBS is denoted by $\text{maxind}(OBS)$.

Note that, since *Props* is finite, for each indicative subset S of a set OBS, there exists at least one maximal indicative subset S' of OBS such that $S \subseteq S'$. Moreover, if OBS is an indicative set of properties itself, there is only one maximal indicative subset of OBS, namely OBS itself.

Based on these notions the following multi-interpretation operator can be defined.

Definition 8.13 (Generic multi-interpretation operator for approximate classification)
For a set $X \subseteq \mathcal{L}_1$, define the *set of observations implied by* X by

$$OBS(X) = \{p \mid \mathsf{observed}(p) \in Cn(X)\}.$$

The *operator* MI_{maxind} is defined by

$$MI_{\mathsf{maxind}}(X) = \{Cn(X \cup KB \cup \{\mathsf{is\_indicative}(p) \mid p \in S\}) \mid S \in \mathrm{maxind}(OBS(X))\}$$

for each $X \subseteq \mathcal{L}_1$.

Note that $X \subseteq Y \subseteq \mathcal{L}_1$ implies $OBS(X) \subseteq OBS(Y)$. Actually, the sets X will often be sets of the form $\{\mathsf{observed}(p) \mid p \in OBS\}$ for some set of properties $OBS \subseteq Props$.

Properties of the generic multi-interpretation operator for approximate classification

The operator MI_{maxind} satisfies a number of properties of well-behavedness. The proofs are rather straightforward.

Proposition 8.14 The multi-interpretation operator MI_{maxind} satisfies inclusion and non-inclusiveness.

In Chapter 10 some further conditions of well-behavedness for belief set operators are introduced (generalizing corresponding properties of inference operations). These properties can be defined for multi-interpretation operators as well; a number of them are formulated below.

Definition 8.15 (Properties of multi-interpretation operators)

1. Let $\mathcal{A}, \mathcal{B}$ be sets of belief sets. The set $\mathcal{B}$ *contains more information than* $\mathcal{A}$, denoted $\mathcal{A} \leq \mathcal{B}$, if for all $T \in \mathcal{B}$ there exists $S \in \mathcal{A}$ such that $S \subseteq T$.

2. Let MI be a multi-interpretation operator. Then MI satisfies *belief monotony* if for all $X, Y \subseteq \mathcal{L}_1$:

$$X \subseteq Y \Rightarrow MI(X) \leq MI(Y).$$

3. Let MI be a multi-interpretation operator for which $\mathcal{L}_1 \subseteq \mathcal{L}_2$.
   (a) MI satisfies *weak belief monotony* if for all $X, Y \subseteq \mathcal{L}_1$:
   $$X \subseteq Y \subseteq K_{MI}(X) \Rightarrow MI(X) \leq MI(Y).$$
   (b) MI satisfies *belief transitivity* if for all $X, Y, T \subseteq \mathcal{L}_1$:
   $$T \in MI(X) \ \& \ X \subseteq Y \subseteq T \Rightarrow K_{MI}(Y) \subseteq T.$$
   (c) MI satisfies *belief cut* if for all $X, Y \subseteq \mathcal{L}_1$:
   $$X \subseteq Y \subseteq K_{MI}(X) \Rightarrow MI(Y) \leq MI(X).$$

Apart from belief monotony (which should in general not be expected), our multi-interpretation operator is well-behaved.

Theorem 8.16 The multi-interpretation operator MI_{maxind} satisfies weak belief monotony, belief transitivity and belief cut. It does not generally satisfy belief monotony.

Proof: Abbreviate MI_{maxind} to MI. Starting with belief monotony, consider a situation in which we have two properties, P_1 and P_2 (for simplicity), and suppose KB contains information which prevents P_1 and P_2 from both being indicative at the same time: there is an attribute A which has possible values 0 and 1. This means that KB contains the formula $\mathsf{has\_value}(A, 0) \vee \mathsf{has\_value}(A, 1)$. Furthermore, suppose that we have $\mathsf{is\_incompatible\_with}(P_1, A, 0)$ and $\mathsf{is\_incompatible\_with}(P_2, A, 1)$ in KB. Now let

$$\begin{array}{rcl} X & = & \{\mathsf{observed}(P_1)\}, \\ Y & = & \{\mathsf{observed}(P_1), \mathsf{observed}(P_2)\}. \end{array}$$

Then $MI(X)$ contains one element (in which P_1 is indicative), and $MI(Y)$ contains two elements, one in which only P_1 is indicative, and one in which only P_2 is indicative. For this latter element there is no smaller set in $MI(X)$. Therefore, belief monotony does not hold.

Let us now consider weak belief monotony and belief cut. Suppose $X \subseteq Y \subseteq K_{MI}(X)$ and let $T \in MI(X)$, then

$$T = Cn(X \cup KB \cup \{\mathsf{is\_indicative}(p) \mid p \in M\})$$

for some $M \in \text{maxind}(OBS(X))$ and $Y \subseteq T$ (since $Y \subseteq K_{MI}(X)$). But as X and Y contain only the predicate $\mathsf{observed}$ which is not present in KB or in $\{\mathsf{is\_indicative}(p) \mid p \in M\}$, it must be the case that $Cn(Y) \subseteq Cn(X)$, so that $Cn(X) = Cn(Y)$. This implies that $MI(X) = MI(Y)$, proving both weak belief monotony and belief cut.

If $T \in MI(X)$ & $X \subseteq Y \subseteq T$, then the same argument shows that $MI(X) = MI(Y)$, from which immediately follows that $K_{MI}(Y) = K_{MI}(X) \subseteq T$. This proves belief transitivity. □

Each of the belief sets is an approximation in the sense of an upper bound of the solution. If the number of observations increases, this upper bound decreases, as is established in the following theorem.

Theorem 8.17 For each pair of subsets $X, Y \subseteq \mathcal{L}_1$ the following holds: $X \subseteq Y \Rightarrow$ for all $S \in MI_{\mathsf{maxind}}(X)$ there exists a $T \in MI_{\mathsf{maxind}}(Y)$ such that $S \subseteq T$.

Proof: From $X \subseteq Y$ it follows $OBS(X) \subseteq OBS(Y)$ (see note just below Definition 8.13). Therefore every maximal indicative subset of $OBS(X)$ is an indicative subset S of $OBS(Y)$. Within $OBS(Y)$ this indicative subset can be extended to a maximal indicative subset S' (see note just below Definition 8.12). This implies the theorem. □

This theorem guarantees that an increasing sequence of observations

$$X_0 \subseteq X_1 \subseteq X_2 \subseteq \ldots$$

results in increasing belief sets within the sets $MI(X_i)$. These increasing belief sets correspond to decreasing sets of classifications, i.e., for each of the increasing belief sets the ranges of the possible values of attributes are decreasing: this provides an approximation of the classification by a sequence of decreasing upper bounds. Note that Theorem 8.17 leaves open the possibility that belief sets remain constant, or new belief sets arise in some stage, i.e., sets of which no subset occurs in the previous set of belief sets. In general, for a given sequence of observations the resulting belief sets will form a set of trees as depicted in Figure 8.5. Here

$$\begin{array}{rcl} MI_{\mathsf{maxind}}(X_0) & = & \{S_0\}, \\ MI_{\mathsf{maxind}}(X_1) & = & \{S_{11}, S_{12}, S_{13}\}, \text{ and} \\ MI_{\mathsf{maxind}}(X_2) & = & \{S_{211}, S_{212}, S_{221}, S_{231}, S_{232}\}. \end{array}$$

The following proposition covers the case of an observed set of properties OBS which has a unique interpretation:

Proposition 8.18 For each subset of properties $OBS \subseteq Props$ the following are equivalent:

1. $MI_{\mathsf{maxind}}(\{\mathsf{observed}(p) \mid p \in OBS\})$ contains just one element.
2. the set OBS is an indicative set of properties.

If these (equivalent) conditions are satisfied, all observed properties are indicative, and there are no alternative interpretations. This means there is no need for

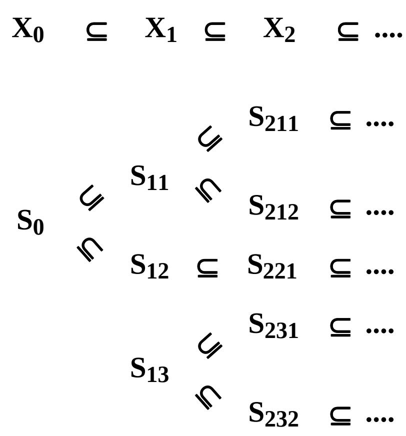

Figure 8.5: Example approximate classifications resulting from an increasing sequence of observations.

further selection from alternatives. The possible values of the attributes are contained in $MI_{\mathsf{maxind}}(\{\mathsf{observed}(p) \mid p \in OBS\})$.

If $MI_{\mathsf{maxind}}(\{\mathsf{observed}(p) \mid p \in OBS\})$ contains more than one element, then a further selection process can be started. But even before this selection process, conclusions can be drawn: the kernel of the MI_{maxind} operator contains the most certain conclusions, so $K_{MI_{\mathsf{maxind}}}(\{\mathsf{observed}(p) \mid p \in OBS\})$ may be inspected. For instance, there may be two possible views in $MI_{\mathsf{maxind}}(\{\mathsf{observed}(p) \mid p \in OBS\})$ due to the fact that there is an attribute A_1 for which no value is compatible with all the observed properties. However, all of these properties may indicate that another attribute A_2 must have a certain value, and this conclusion will be in $K_{MI_{\mathsf{maxind}}}(\{\mathsf{observed}(p) \mid p \in OBS\})$. If A_2 is all one is interested in, there is no need for selection. If one is interested also in A_1, this selection has to take place. If one is interested in the properties which are indicative in both maximal indicative sets, one can either examine $K_{MI_{\mathsf{maxind}}}(\{\mathsf{observed}(p) \mid p \in OBS\})$, or the intersection of the maximal indicative sets:

$$K_{MI_{\mathsf{maxind}}}(X) \cap \{\mathsf{is\_indicative}(p) \mid p \in P\} =$$

$$\{\mathsf{is\_indicative}(p) \mid p \in \bigcap \mathrm{maxind}(OBS(X))\}.$$

For the multi-interpretation operator MI_{maxind}, the language and the format (the kinds of rules) of the knowledge base KB were fixed. When the language and format of KB is left open, we get a general class of multi-interpretation operators that can deal with input which is contradictory in the sense that it is inconsistent with a knowledge base.

8.2.2 Representation in default logic

The previous subsection described the generic multi-interpretation operator MI, which formalizes the interpretation of (possibly inconsistent) observation information using maximal indicative sets. A specification of this multi-interpretation operator in a (well-known) logical formalism would mean that known results about this logic can be applied to this situation, but it would also allow for the use of proof mechanisms for this logic to be used in an implemented system based on such an operator. In Section 7.2 and in [MTT97] default logic is used as a specification language for belief frames. These results can be applied to the formalization of the previous subsection.

In [MTT97] the following theorem was proven (Corollary 5.2):

Theorem 8.19 A family F of theories is representable by a *normal* default theory if and only if $F = \{\mathcal{L}\}$ or there is a consistent set of formulae W and a set of formulae C such that

$$F = \{\,Cn(W \cup \Phi) \mid \Phi \text{ is a maximal subset of } C \text{ consistent with } W\}.$$

In Section 7.2 the question is posed whether a multiple belief state operator in IS^{syn} can be represented by a set of defaults. Below, the definition of representability is slightly generalized to deal with a different input and output language. Also, we allow some axioms besides the input facts. Recall that $\mathcal{L}_1$ is the input language, and $\mathcal{L}_2$ is the output language. We make the assumption that $\mathcal{L}_1 \subseteq \mathcal{L}_2$.

Definition 8.20 (Representability of a multi-interpretation operator) Let $\Delta = \langle D, W\rangle$ be a default theory. A multi-interpretation operator MI is *representable by* Δ, if for all $X \subseteq \mathcal{L}_1$ it holds that $MI(X) = \mathsf{Ext}(\langle D, W \cup X\rangle)$. The operator MI is called *representable by a default theory* if there exists such a default theory.

Consider the family of belief sets $MI_{\mathsf{maxind}}(X)$ where $X \subseteq \mathcal{L}_1$. Then Theorem 8.19 can be applied to $MI_{\mathsf{maxind}}(X)$ by setting:

$$\begin{aligned} W &= X \cup KB, \text{ and} \\ C &= \{\mathsf{is\_indicative}(p) \mid p \in OBS(X)\}. \end{aligned}$$

Therefore Theorem 8.19 implies that for each $X \subseteq \mathcal{L}_1$ there exists a normal default theory that represents the belief sets of $MI_{\mathsf{maxind}}(X)$. The theorem does not imply that there exists *one* set of defaults D which works for all sets $X \subseteq \mathcal{L}_1$, so this does not imply that the multi-interpretation operator MI_{maxind} is representable by a default theory. However, the normal default theory can actually be found by defining the following generic set of defaults D:

$$\frac{\mathsf{observed}(p) : \mathsf{is\_indicative}(p)}{\mathsf{is\_indicative}(p)} \qquad \text{for all properties } p \in Props.$$

This set of defaults is independent of X, so MI_{maxind} is representable.

Theorem 8.21 The multi-interpretation operator MI_{maxind} is representable by the normal default theory $\langle D, KB\rangle$.

Proof: Let X be a set of formulae in $\mathcal{L}_1$. Let $KB \cup X$ be consistent (if it is not, verification is straightforward and omitted). The extensions of $\langle D, KB \cup X\rangle$ are sets of the form $Cn(KB \cup X \cup S)$, where S is a subset of $\{\mathsf{is\_indicative}(p) \mid \mathsf{observed}(p) \in Cn(X)\}$, which is maximal such that $Cn(KB \cup X \cup S)$ is consistent. This is proved below. The sets $Cn(KB \cup X \cup S)$ with S as above together comprise $MI_{\mathsf{maxind}}(X)$.

First of all, let S be such a maximal set, and let $E = Cn(KB \cup X \cup S)$. Then if the E_i are defined as in Definition 3.1, the following holds:

$$
\begin{array}{lcl}
E_0 & = & Cn(KB \cup X), \text{ and} \\
E_1 & = & Cn(E_0 \cup \{\mathsf{is\_indicative}(p) \mid \mathsf{observed}(p) \in E_0, \neg\mathsf{is\_indicative}(p) \notin E\}).
\end{array}
$$

As E_1 does not contain more instances of the observed predicate than E_0 (this follows from the fact that X contains only the observed predicate, whereas KB does not), $E_i = E_1$ for all $i > 1$. The claim is that

$$\{\mathsf{is\_indicative}(p) \mid \mathsf{observed}(p) \in E_0, \neg\mathsf{is\_indicative}(p) \notin E\} = S.$$

Suppose $\mathsf{observed}(p) \in E_0$ and $\neg\mathsf{is\_indicative}(p) \notin E$. Then $\mathsf{observed}(p)$ is in $Cn(X)$ and $Cn(KB \cup X \cup S \cup \{\mathsf{is\_indicative}(p)\})$ is consistent. But as S was maximal with respect to these properties, $\mathsf{is\_indicative}(p) \in S$. On the other hand, if $\mathsf{is\_indicative}(p) \in S$, then $\mathsf{observed}(p) \in E_0$ and $\neg\mathsf{is\_indicative}(p) \notin E$ (as $E = Cn(KB \cup X \cup S)$ is consistent).

Now let E be an extension of $\langle D, KB \cup X\rangle$. Then E is of the form $Cn(KB \cup X \cup S)$, where S contains (only) formulae of the form $\mathsf{is\_indicative}(p)$. Examination of KB (and the fact that $X \subseteq \mathcal{L}_1$), shows that only if $\mathsf{observed}(p) \in Cn(X)$ is $\mathsf{is\_indicative}(p) \in E$. As extensions are always consistent (if each rule has a justification and the axioms are consistent), $Cn(KB \cup X \cup S)$ must be consistent. Suppose there exists a $T \supset S$ (strict inclusion) respecting the conditions, then there must be a default rule $(\mathsf{observed}(p) : \mathsf{is\_indicative}(p))/\mathsf{is\_indicative}(p)$, with $\mathsf{observed}(p) \in Cn(X) \subseteq E$ and $Cn(KB \cup X \cup S \cup \{\mathsf{is\_indicative}(p)\})$ consistent, implying that $\neg\mathsf{is\_indicative}(p) \notin E$. But that means there is an applicable default rule for which the conclusion is not in E, contradicting the assumption that E is an extension. Therefore S must be maximal. □

At this point the reader may wonder what the benefit is of the representation in default logic. The multi-interpretation operator MI_{maxind} arose during the analysis and formalization of an application to be described in the next section. A system, EKS, was implemented based on this operator MI_{maxind}. The implementation in fact follows the definition (Definition 8.13) rather closely. The results of the current section indicate that alternatively a theorem prover for default logic (or, rather, a program computing extensions of default theories) could be used. A highly optimized theorem prover for default logic obviates the need to optimize this part of the system ourselves. This is one of the subjects of current work on the system.

8.2.3 Application: EKS

In this section we will briefly describe a domain to which the formalization above was applied (see also [BET98]). Nature conservationists are interested in a number of so-called *abiotic factors* of terrains. These factors, examples of which are the moisture, acidity and nutrient value, give an indication of how healthy a terrain is. As these factors are difficult to measure directly, a sample of plant species growing on a terrain is taken. For each species, the experts have knowledge about the possible values of the abiotic factors of a terrain on which the species lives. An example of such a sample, together with the knowledge about the values of abiotic factors for each plant in the sample, is given in Table 8.1. Combining such knowledge for

	Moisture						Acidity					Nutrient Value			
Species	vd	fd	fm	vm	fw	vw	bas	neu	sac	fac	ac	np	fnr	nr	vnr
Angelica sylvestris				x	x		x	x					x	x	
Caltha palustris ssp palustris				x	x		x	x	x			x	x	x	
Carex acutiformis				x	x		x	x					x	x	
Carex acuta				x	x	x	x	x	x				x	x	x
Deschampsia caespitosa			x	x	x		x	x	x				x	x	x
Epilobium parviflorum			x	x			x	x	x				x	x	
Equisetum palustre			x	x	x	x	x	x	x			x	x	x	
Galium palustre				x	x		x	x	x			x	x	x	x
Glyceria fluitans				x	x	x	x	x	x	x			x	x	x
Juncus articulatus				x	x		x	x	x			x	x	x	x
Lathyrus pratensis			x	x			x	x	x				x	x	
Myosotis palustris				x	x		x	x	x				x	x	x
Phalaris arundinacea			x	x	x	x	x	x						x	x
Phleum pratense ssp pratense			x	x			x	x						x	x
Poa trivialis			x	x	x		x	x						x	x
Scirpus sylvaticus				x	x	x	x	x	x				x	x	

Moisture (vd: very dry, fd: fairly dry, fm: fairly moist, vm: very moist, fw: fairly wet, vw: very wet), Acidity (bas: basic, neu: neutral, sac: slightly acid, fac: fairly acid, ac: acid), Nutrient value (np: nutrient poor, fnr: fairly nutrient rich, nr: nutrient rich, vnr: very nutrient rich)

Table 8.1: Example sample.

each of the plant species observed on a terrain leads to conclusions about the abiotic factors of the terrain. If we look, for example, at *Caltha palustris* L., then we see that the terrain must be:

- very moist or fairly wet,
- basic, neutral or slightly acid, and
- nutrient poor, fairly nutrient rich or nutrient rich.

For the species *Poa trivialis* L. the terrain needs to be

- fairly moist, very moist or fairly wet,
- basic or neutral, and
- nutrient rich or very nutrient rich.

When both species occur in a terrain, this implies that the terrain can only be:

- very moist or fairly wet,
- basic or neutral, and
- nutrient rich.

Analysis of the abiotic conditions for all plant species presented in Table 8.1 shows that only a restricted number of possibilities (but more than one) for the abiotic conditions can be found in which all of these plant species can abide. This *greatest common denominator* for the given plant species is defined by the following set of abiotic conditions:

- very moist,
- basic or neutral, and
- nutrient rich.

The combination of these plant species indicates that a terrain on which these plant species are found has to fulfill these conditions.

During the development of a knowledge-based system, EKS to automate this classification process, however, it turned out that the samples of species taken were often incompatible. That is, there was at least one abiotic factor for which no value could be found that was permissible for all species. An example of such a sample (taken at the Pommeren site), is presented in Table 8.2. For this sample, there is no common denominator of abiotic conditions. Focusing on the acidity of a terrain shows that the plant species *Angelica sylvestris* L., for example, only grows on a basic or neutral terrain, whereas the species *Carex panicea* L., also found in the same sample, only grows on a slightly or fairly acid terrain. These two species, however, are in the same sample. One common set of possible values of the abiotic factors for all plant species can not be derived. This is not due to errors in the knowledge of abiotic factors needed by species to live, but due to other effects. For example, a terrain may lie on the transition of a dry and a wet piece of land. Some of the observed species may occur on the drier, and others on the wetter side. This can also be due to the presence of ponds in an otherwise dry terrain. Also transitions of a terrain over time, or vertical inhomogeneity may be causes.

The approximate classification task described above is an example of the task performed by the multi-interpretation operator MI_{maxind}. The object to be studied is a terrain, and the attributes of interest are the abiotic factors. The presence of certain species are the observable properties of the object. So we can specialize the

	Moisture						Acidity					Nutrient Value			
Species	vd	fd	fm	vm	fw	vw	bas	neu	sac	fac	ac	np	fnr	nr	vnr
Angelica sylvestris				x	x		x	x					x	x	
Anthoxanthum odoratum		x	x	x					x	x		x	x		
Caltha palustris ssp palustris				x	x		x	x	x			x	x	x	
Carex acutiformis				x	x		x	x					x	x	
Carex acuta				x	x	x	x	x	x				x	x	x
Carex nigra			x	x	x				x	x	x	x	x		
Carex panicea			x	x	x				x	x		x	x		
Carex riparia				x	x	x	x	x						x	x
Cirsium oleraceum				x	x		x	x					x	x	
Cirsium palustre				x			x	x	x			x	x	x	
Crepis paludosa			x	x	x		x	x	x				x	x	
Deschampsia caespitosa			x	x	x		x	x	x				x	x	x
Epilobium palustre			x	x	x				x			x	x		
Epilobium parviflorum			x	x			x	x	x				x	x	
Equisetum palustre			x	x	x	x	x	x	x			x	x	x	
Filipendula ulmaria				x			x	x	x			x	x	x	
Galium palustre				x	x		x	x	x			x	x	x	x
Glyceria fluitans				x	x	x	x	x	x	x			x	x	x
Juncus articulatus				x	x		x	x	x			x	x	x	x
Juncus conglomeratus		x	x	x					x	x		x	x		
Lathyrus pratensis			x	x			x	x	x				x	x	
Lotus uliginosus			x	x	x		x	x	x			x	x	x	
Lychnis flos cuculi				x	x		x	x	x				x	x	
Lysimachia vulgaris			x	x	x		x	x	x			x	x	x	
Myosotis palustris				x	x		x	x	x				x	x	x
Phalaris arundinacea			x	x	x	x	x	x						x	x
Phleum pratense ssp pratense			x	x			x	x						x	x
Poa trivialis			x	x	x		x	x						x	x
Scirpus sylvaticus				x	x	x	x	x	x				x	x	

Moisture (vd: very dry, fd: fairly dry, fm: fairly moist, vm: very moist, fw: fairly wet, vw: very wet), Acidity (bas: basic, neu: neutral, sac: slightly acid, fac: fairly acid, ac: acid), Nutrient value (np: nutrient poor, fnr: fairly nutrient rich, nr: nutrient rich, vnr: very nutrient rich)

Table 8.2: Second example sample.

generic knowledge base to this case. The language is as follows:

properties: (occurrence of) plant species	achillea_millefolium, achillea_ptarmica, ...
attributes: abiotic factors	moisture, acidity, nutrient_value
values for each of the attributes: (abiotic factors)	very_dry, fairly_dry, ..., basic, neutral, ..., nutrient_poor, fairly_nutrient_rich, ...

The experts' knowledge about the possible values of abiotic factors for a species (as presented in the Tables 8.1 and 8.2 above), is formalized by (a large number of) instances of the predicate is_incompatible_with(P, A, V), where P is one of the plant species names, A is an abiotic factor, and V is a value of that factor. This knowledge leads to a specific instantiation of MI_{maxind}, which we will denote by the same name. The alternative interpretations given by $MI_{\mathsf{maxind}}(X)$ are extremely useful. Each of the alternatives leads to a different set of (possible) values for the abiotic factors. The two alternatives created for the second example sample, are indicated by ovals in Table 8.3. If this is for instance due to the fact that the terrain consists of a drier portion and a wetter portion, then a selection can be made for the portion of interest, whose possible values for abiotic factors are contained in the corresponding interpretation. This selection process can be formalized by a selection operator. At this moment, that process has not been analyzed in more detail, but that is one of the future directions of research.

As mentioned before, a system called EKS has been developed, using the environment DESIRE described earlier, to help a user in establishing the abiotic factors of a terrain. The correspondence between the formalization of the expert reasoning task and the interactive knowledge-based system EKS that models the approximate classification task is as follows. The first component of the system, determination_of_maximal_indicative_ subsets, is formalized by the belief set operator MI_{maxind} defined earlier. The second component of the system,
selection_of_a_maximal_indicative_subset, which models (an interface to) the selection process by the user of the system, is formalized by a single-valued selection function s_{user}.

The composition C_{EKS} of MI_{maxind} and s_{user} defined by

$$C_{\mathsf{EKS}}(X) = s_{\text{user}}(MI_{\mathsf{maxind}}(X)) \text{ for } X \subseteq \mathcal{L}_1$$

is a selective interpretation operator for MI_{maxind} (as described in Definition 8.11; see also the remark immediately following the definition). This operator formalizes the reasoning of the system in interaction with the user. Note that from the two functions of which this overall function is composed, one is fixed and defined by the system itself (i.e., MI_{maxind}), whereas the other can be changed dynamically, depending on the user (i.e., s_{user}). For more details on this application, see [BET98].

8.2.4 Conclusions and related work

In most real-life classification problems, the information about the object to be classified can be interpreted in different ways. In this section, multi-interpretation operators were introduced to formalize this interpretation process. In particular, observation results of the world may underspecify or overspecify a classification. Overspecification means that the observations are in contradiction with knowledge about the world. A generic multi-interpretation operator was introduced for approximate classification tasks where attribute values of an object are determined on the basis of imperfect interpretation of observable properties of the object. The multi-interpretation operator formalizes in an integrated fashion the different variants of approximate classifications of the object. This operator is rather well-behaved, and

	Moisture						Acidity					Nutrient Value			
Species	vd	fd	fm	vm	fw	vw	bas	neu	sac	fac	ac	np	fnr	nr	vnr
Angelica sylvestris				x	x		x	x					x	x	
Carex acutiformis				x	x		x	x					x	x	
Carex riparia				x	x	x	x	x						x	x
Cirsium oleraceum				x	x		x	x					x	x	
Phalaris arundinacea			x	x	x	x	x	x						x	x
Phleum pratense ssp pratense			x	x			x	x						x	x
Poa trivialis			x	x	x		x	x						x	x
Caltha palustris ssp palustris				x	x		x	x	x			x	x	x	
Carex acuta				x	x	x	x	x	x				x	x	x
Cirsium palustre				x			x	x	x			x	x	x	
Crepis paludosa			x	x	x		x	x	x				x	x	
Deschampsia caespitosa			x	x	x		x	x	x				x	x	x
Epilobium parviflorum			x	x			x	x	x				x	x	
Equisetum palustre			x	x	x	x	x	x	x			x	x	x	
Filipendula ulmaria				x			x	x	x			x	x	x	
Galium palustre				x	x		x	x	x			x	x	x	x
Glyceria fluitans				x	x	x	x	x	x	x			x	x	x
Juncus articulatus				x	x		x	x	x			x	x	x	x
Lathyrus pratensis			x	x			x	x	x				x	x	
Lotus uliginosus			x	x	x		x	x	x			x	x	x	
Lychnis flos cuculi				x	x		x	x	x				x	x	
Lysimachia vulgaris			x	x	x		x	x	x			x	x	x	
Myosotis palustris				x	x		x	x	x				x	x	x
Scirpus sylvaticus				x	x	x	x	x	x				x	x	
Anthoxanthum odoratum		x	x	x					x	x		x	x		
Carex nigra			x	x	x				x	x	x	x	x		
Carex panicea			x	x	x				x	x		x	x		
Epilobium palustre			x	x	x				x			x	x		
Juncus conglomeratus		x	x	x					x	x		x	x		

Moisture (vd: very dry, fd: fairly dry, fm: fairly moist, vm: very moist, fw: fairly wet, vw: very wet), Acidity (bas: basic, neu: neutral, sac: slightly acid, fac: fairly acid, ac: acid), Nutrient value (np: nutrient poor, fnr: fairly nutrient rich, nr: nutrient rich, vnr: very nutrient rich)

Table 8.3: Maximal indicative subsets for the second example sample.

can be represented by a default theory. This can be a basis for the use of (highly optimized) theorem provers for default logic, to implement a system formalized by the multi-interpretation operator. For the domain of ecological classification an application for the theory has been developed, and the resulting system, EKS, that has been implemented has shown great promise to be a useful tool for nature conservationists.

Classification tasks form an important application area for knowledge-based systems (see for example [Ste95]). The phenomenon of underspecification leading to more than one candidate solution is ubiquitous and handled by any approach to classification. The phenomenon of overspecification as a result of conflicting information from different sensors is also known, and often handled by some numeric method of sensor information fusion. As far as we know, our approach is the first integrated qualitative treatment of classification with both kinds of phenomena.

After multiple interpretations of observation information have been identified, often a choice is made for one of them. Which view is (or which views are) most appropriate presumably requires additional heuristic (strategic) knowledge (cf. [Bre94a], [Bre94b], [TT92]). One of the areas of future research is to further analyze this choice process, in general terms, but also in particular for the knowledge-based system. Future research will focus on the acquisition of this knowledge to be able to support users in the selection process.

Acknowledgments

Some of the material from Section 8.1 appeared in [EJT98], whereas the material in Section 8.2 is described in [ET98].

Chapter 9

Some Logical Themes

In this chapter, we will treat a number of items linked with the temporal logics of information described in Chapter 4. The first section looks at logical properties of MTEL, such as decidability, complexity and axiomatizability. In the next section, classes of formulae are studied under which the consequence relation behaves monotonically, not only for MTEL, but for preferential logics in general (MTEL falls into this category). Lastly, the consequence relation of MTEL is not cumulative and therefore does not belong to the class of relations studied in [KLM90]. Given our interest in this logic, we develop a framework analogous to the one of [KLM90], but for consequence relations that do not necessarily satisfy the rule of Cautious Monotonicity.

9.1 Axioms, decidability and complexity of MTEL

We begin this section by looking at a proof system for temporal epistemic logic.

9.1.1 Axiom systems for TEL and TELC

As TEL can be seen as a logical combination of S5 and tense logic over the natural numbers, we shall combine the axioms of both logics in order to obtain a proof system for TEL. Instead of proving soundness and completeness for the resulting system from scratch, we will use results from [FG92] (see also [FG96]) where a general method for temporalizing a given logic system is presented. In their notation, TEL would be T(S5)). We cannot directly apply their results since they use the temporal operators Since and Until, but adaptation to our situation is easy.

Axiomatizations for S5 are known from the literature (e.g. [HM85a]):

Definition 9.1 (Axiom system for S5) The axiom system of **S5** consists of:

1. All instances of propositional tautologies
2. $K(\varphi \to \psi) \to (K\varphi \to K\psi)$ (K)
3. $K\varphi \to \varphi$ (T)
4. $K\varphi \to KK\varphi$ (*Positive Introspection*)
5. $\neg K\varphi \to K\neg K\varphi$ (*Negative Introspection*)

and the following two rules:

1. $\dfrac{\varphi \quad \varphi \to \psi}{\psi}$ (*Modus Ponens*)
2. $\dfrac{\varphi}{K\varphi}$ (*Necessitation*)

If there is a proof for φ using this system, we will denote this by $\vdash_{\mathbf{S5}} \varphi$.

It is well-known that this system is sound and complete with respect to the class of normal S5-models (see for instance [MH95]).

The results in [FG92] concern temporalizing a logic over a class of flows of time. Our class of flows of time contains only the set of natural numbers. First we will give an axiomatic system for propositional tense logic over the natural numbers (from [Gol92]), which is sound and complete with respect to $\mathbb{N}$:

Definition 9.2 (Tense logic over the natural numbers) The axiom system for tense logic over $\mathbb{N}$ consists of:

1. All instances of propositional tautologies
2. $G(\varphi \to \psi) \to (G\varphi \to G\psi)$
3. $H(\varphi \to \psi) \to (H\varphi \to H\psi)$
4. $\varphi \to HF\varphi$ (C_P)
5. $\varphi \to GP\varphi$ (C_F)
6. $H\varphi \to HH\varphi$ (4_P)
7. $G\varphi \to GG\varphi$ (4_F)
8. $F(\top)$ (D_F)
9. $G(G\varphi \to \varphi) \to (FG\varphi \to G\varphi)$ (Z_F)
10. $H(H\varphi \to \varphi) \to H\varphi$ (W_P)

and the following rules:

1. $\dfrac{\varphi \quad \varphi \to \psi}{\psi}$ (*Modus Ponens*)
2. $\dfrac{\varphi}{G\varphi} \qquad \dfrac{\varphi}{H\varphi}$ (*Necessitation*)

Using the axiom systems for S5 and tense logic, Definition 2.6 of [FG92] allows us to give an axiomatization for TEL:

Definition 9.3 (Axiomatization for TEL) The axiom system of **TEL** consists of:

1. The axioms 1-10 of Definition 9.2.

2. The inference rules 1 and 2 of Definition 9.2.

3. For every formula $\alpha \in \mathcal{L}_{S5}$, if $\vdash_{\mathbf{S5}} \alpha$ then $\vdash_{\mathbf{TEL}} \alpha$ (*Preserve*).

Using Theorem 2.2 of [FG92], soundness of **S5** and soundness of the axiom system for tense logic over $\mathbb{N}$, we immediately have:

Theorem 9.4 (Soundness of TEL) The axiom system **TEL** is sound: for every $\varphi \in \mathcal{L}_{TEL}$ it holds

$$\models \varphi \ \Rightarrow \vdash_{\mathbf{TEL}} \varphi.$$

Theorem 2.3 of [FG92] states that if the system to be temporalized is complete and the axiomatization of the logic with Since and Until is complete over a class of *linear* flows of time, then the 'merged' axiomatization is complete for the temporalized logic. Our class of flows of time (consisting only of the natural numbers) is a subclass of the linear flows of time. A slight adaptation of their proof yields the same result for temporalizing over the temporal operators used in TEL. Therefore we have:

Theorem 9.5 (Completeness of TEL) The axiom system **TEL** is complete: for every $\varphi \in \mathcal{L}_{TEL}$ it holds

$$\vdash_{\mathbf{TEL}} \varphi \ \Rightarrow \models \varphi.$$

Again borrowing from [FG92], Theorem 3.1, and using the fact that both S5 (see [MH95]) and tense logic over the natural numbers (see [SC85]) are decidable, we have:

Theorem 9.6 (Decidability of TEL) The logic TEL is decidable.

Remember that the logic TELC was formed by restricting attention to conservative TEL-models (see Definition 4.6). We can easily find an axiom system for TELC by adding axioms for conservativity.

Proposition 9.7 (Axiomatization of TELC) Let **Cons** $= \{\Box(K\alpha \to G(K\alpha)) \mid \alpha$ a propositional formula $\}$. For each closed TEL-model $\mathcal{M}$ the following are equivalent:

1. $\mathcal{M}$ is conservative,
2. $\mathcal{M} \models$ **Cons**,
3. $(\mathcal{M}, t) \models$ **Cons** for some $t \in \mathbb{N}$.

Furthermore, the axiom system **TELC**, consisting of **TEL** plus the axioms of **Cons**, is sound and complete with respect to the class of TELC-models: for every $\varphi \in \mathcal{L}_{TEL}$ it holds:

$$\models^c \varphi \;\Leftrightarrow \vdash_{\mathbf{TELC}} \varphi.$$

Proof: Let $\mathcal{M}$ be a conservative TEL-model, not necessarily closed, and let $t \in \mathbb{N}$. Suppose $(\mathcal{M}, t) \models K\alpha$ and take $s > t$ arbitrary. Then for all $m \in \mathcal{M}_t$, $m \models \alpha$. Take $m \in \mathcal{M}_s$, then since $\mathcal{M}$ is conservative we have $\mathcal{M}_s \subseteq \mathcal{M}_t$, so $m \in \mathcal{M}_t$ and $m \models \alpha$. Therefore $(\mathcal{M}, s) \models K\alpha$, and since s was arbitrary we have $(\mathcal{M}, t) \models G(K\alpha)$, so $(\mathcal{M}, t) \models K\alpha \to G(K\alpha)$. As t was arbitrary, we have that for any $s \in \mathbb{N}$ it holds that $(\mathcal{M}, s) \models \Box(K\alpha \to G(K\alpha))$. This shows that statement 1 implies statement 2, and also that **TELC** is sound with respect to the class of conservative models.

Now let $\mathcal{M}$ be a closed TEL-model. Suppose that $(\mathcal{M}, t) \models$ **Cons** for some $t \in \mathbb{N}$, but $\mathcal{M}$ is not conservative. Then there exists $s \in \mathbb{N}$ such that $\mathcal{M}_s \not\subseteq \mathcal{M}_{s+1}$. As $\mathcal{M}$ is closed, this means that there must exist a propositional formula α such that $(\mathcal{M}, s) \models K\alpha$ whereas $(\mathcal{M}, s+1) \not\models K\alpha$. Thus $(\mathcal{M}, t) \not\models \Box(K\alpha \to G(K\alpha))$, a contradiction. This means that statement 3 implies statement 1. It is immediate that statement 2 implies statement 3.

The only thing left to prove is completeness, so suppose $\models^c \varphi$. As the notion $\models^c$ is independent of the (propositional) signature P (a fact we will prove later on, see Proposition 9.9), we take it finite (actually, we can restrict it to the propositional atoms occurring in φ). So we have for all TEL-models $\mathcal{M}$: if $\mathcal{M}$ is conservative then $\mathcal{M} \models \varphi$. Since there are only a finite number of non-equivalent propositional formulae over P, we can take **Cons** to be finite and therefore we can take the conjunction of its elements. So if $(\mathcal{M}, s) \models \bigwedge$**Cons** then $\mathcal{M}$ is conservative (by the first part of this proposition; $\mathcal{M}$ is closed as all models are closed when the signature is finite), so $\mathcal{M} \models \varphi$ and therefore $(\mathcal{M}, s) \models \varphi$. Thus we have $\bigwedge$**Cons** $\models \varphi$, and using the deduction lemma for TEL (which can be easily verified), $\models \bigwedge$**Cons** $\to \varphi$, from which by the completeness of **TEL** (Theorem 9.5) it follows that $\vdash_{\mathbf{TEL}} \bigwedge$**Cons** $\to \varphi$. Since **TELC** contains **TEL** and the axioms of **Cons**, and has Modus Ponens as inference rule, we conclude $\vdash_{\mathbf{TELC}} \varphi$. $\Box$

We also have that TELC is decidable:

Proposition 9.8 (Decidability of TELC) The logic TELC is decidable.

Proof: Checking whether $\vdash_{\mathbf{TELC}} \varphi$ reduces to checking $\vdash_{\mathbf{TEL}} \bigwedge C \to \varphi$ where C is the set of rules $\Box(K\alpha \to G(K\alpha))$ for all non-equivalent propositional formulae α in the proposition letters of φ. This is decidable by Theorem 9.6. $\Box$

We conclude this section with some other logical characteristics of TEL and its derived logics.

The notions $\models$ (of TEL) and $\models^c$ are not compact: the set $\{P^i(\top) \mid i \in \mathbb{N}\}$ (where P^i stands for a sequence of P operators of length i) is not satisfiable, whereas each finite subset is (for both notions).

Both notions above, as well as consequence in MTEL, are independent of the signature. From now on, we will consider entailment in MTEL only as a relation between formulae, instead of a relation between sets of formulae and formulae. Equivalently, we could restrict the set T in Definition 4.45 to be a finite set.

Proposition 9.9 The notions $\models$, $\models^c$ and $\models_{\preceq^g}$ are independent of the propositional signature P.

Proof: For a propositional signature P we write $\mathcal{L}_P$ to denote the temporal epistemic language based on P and P-$\models$, P-$\models^c$, P-$\models_{\preceq^g}$ to denote the corresponding relations based on this signature.

In the proof, we will need a number of constructions. Let P, Q be two propositional signatures with $P \subseteq Q$ (note that we can make the assumption that $P \subseteq Q$ without loss of generality) . For a propositional valuation m of signature Q, let $m|_P$ denote the restriction of m to atoms of P. Now define:

- For a TEL-model $\mathcal{M}$ based on Q, we define its restriction to P, denoted $\mathcal{M}|_P$, by:

$$(\mathcal{M}|_P)_s = \{m|_P \colon m \in \mathcal{M}_s\}.$$

- For a TEL-model $\mathcal{M}$ based on P, we define its extension to Q, denoted $\mathcal{M}|^Q$, by:

$$(\mathcal{M}|^Q)_s = \{m \in \mathrm{Val}(Q) : m|_P \in \mathcal{M}_s\}.$$

A straightforward induction on $\varphi \in \mathcal{L}_P$ shows that truth of φ at a point in time is preserved under these constructions.

Independence of the signature for $\models$ is now straightforward since any countermodel that denies P-$\models \varphi$ can be transformed to a countermodel which denies Q-$\models \varphi$ and vice versa. The same holds for $\models^c$ since both constructions above preserve conservativity.

For $\models_{\preceq^g}$, suppose that $\mathcal{M}$ is a conservative TEL-model based on Q and $\mathcal{M}$ is a $\preceq^g$-minimal model of φ (with $\preceq^g$ defined over models of signature Q). Then $\mathcal{M}|_P$ is a $\preceq^g$-minimal of φ (with $\preceq^g$ defined over models of signature P): for suppose $\mathcal{N}$

is a conservative TEL-model based on P with $\mathcal{N} \prec^g \mathcal{M}|_P$ and $\mathcal{N} \models \varphi$, then (!) $\mathcal{N}|^Q \prec^g \mathcal{M}$ and $\mathcal{N}|^Q \models \varphi$.

Conversely, suppose that $\mathcal{M}$ is a conservative TEL-model based on P and $\mathcal{M}$ is a $\preceq^g$-minimal model of φ (with $\preceq^g$ defined over models of signature P). Then $\mathcal{M}|^Q$ is a $\preceq^g$-minimal of φ (with $\preceq^g$ defined over models of signature Q) : for suppose $\mathcal{N}$ is a conservative TEL-model based on Q with $\mathcal{N} \prec^g \mathcal{M}|^Q$ and $\mathcal{N} \models \varphi$, then (!) $\mathcal{N}|_P \prec^g \mathcal{M}$ and $\mathcal{N}|_P \models \varphi$.

It is now easy to see that φ P-$\models_{\preceq^g} \psi$ if and only if φ Q-$\models_{\preceq^g} \psi$. □

We showed that both TEL and TELC have sound and complete (Hilbert-style) proof systems, and that they are decidable. These same issues are also of interest for MTEL. The next section treats decidability of MTEL, and complexity of MTEL (and TEL) are the subject of the section thereafter. As far as a proof system for MTEL is concerned, this seems to be much harder to find. In general, proof systems for nonmonotonic logics are hard to find (proof systems for default logic were proposed only recently: [Bon96] proposes a sequent calculus for credulous entailment, and [BO97] contains a calculus for skeptical entailment). Instead of giving a direct proof system, another possibility is to embed (or translate) minimal entailment into another logic, for which there is a proof system. We will briefly sketch such an approach.

Consider the set of TELC-models (which we denote by $\mathcal{TELC}$) with the relation $\prec^g$. This can be seen as a pair, with a set of states and an accessibility relation. This relation (or its converse) can be used to interpret a modal operator $\blacklozenge$ in the standard way:

$$(\mathcal{TELC}, \mathcal{M}) \models \blacklozenge\varphi \;\Leftrightarrow\; \text{there is } \mathcal{N} \in \mathcal{TELC} \text{ with } \mathcal{N} \prec^g \mathcal{M} \text{ and } (\mathcal{TELC}, \mathcal{N}) \models \varphi.$$

For φ which do not contain this new modal operator, we can define

$$(\mathcal{TELC}, \mathcal{M}) \models \varphi \quad \Leftrightarrow \quad \mathcal{M} \models \varphi.$$

This defines a modal logic 'over' TELC (whereas usually, modal logics are defined over propositional logic or first-order predicate logic). A formula is a theorem in this logic if it is true in all states in the above 'super-model' $\mathcal{TELC}$. We can translate entailment in MTEL in this new logic: we have that $\varphi \models_{\preceq^g} \psi$ exactly when ψ holds in all minimal models of φ. A minimal model of φ is a model in which φ holds, but for which there is no $\preceq^g$ smaller model in which φ holds. So $\varphi \models_{\preceq^g} \psi$ holds exactly when the following formula is a theorem in this new modal logic:

$$(\varphi \wedge \neg\blacklozenge\varphi) \rightarrow \psi.$$

It is easy to see that this logic is monotonous (or actually, the consequence relation $\models_{Smtel}$ defined by $\alpha \models_{Smtel} \beta$ if and only if $\alpha \rightarrow \beta$ is a theorem), so finding a proof system for it may be easier. Such an approach was successfully tried for the 'easier' logic Ground S5, which is essentially the non-temporal variant of MTEL (see [EV98]). Ground S5 is a slight generalization of Halpern & Moses logic of 'Only Knowing' (see [HM85b]). A definition of Ground S5 can be found in Section 9.2.2.

We are interested in the complexity of minimal entailment in MTEL; we will first concentrate on the decidability.

9.1.2 Decidability of MTEL

The first question to be asked when investigating the complexity of a notion is whether it is decidable or not. The notion of minimal entailment of MTEL will turn out to be decidable, but in order to prove that we will first need some lemmas. As the notion $\models_{\preceq^g}$ is independent of the propositional signature, when deciding whether $\varphi \models_{\preceq^g} \psi$, we may take the signature to consist of the atoms occurring in φ and ψ. Therefore, some of the results below are proved for finite signatures.

Observation 9.10 Suppose the signature P is finite. A conservative TEL-model $\mathcal{M}$ consists of a sequence of normal S5-models. These models consist of a finite number of propositional valuations, since P is assumed to be finite. Furthermore the sequence is (not necessarily strictly) decreasing. Therefore there must exist a time point $s \in \mathbb{N}$ such that for all $t > s : \mathcal{M}_t = \mathcal{M}_s$. If s_0 is the smallest point for which this is true, we say that $\mathcal{M}$ *stabilizes* at s_0.

Since all TELC-models stabilize when the signature is finite, it is possible to store them in finite space.

The idea in the proof of decidability is that for each formula ψ there is a number n_ψ such that a minimal model of ψ must stabilize before n_ψ. Then there is only a finite number of models to be checked (taking P finite), and since they stabilize, it is always possible to check whether a temporal formula holds in them. To obtain the upper bound n_ψ one reasons that if there exists a long enough sequence of identical states in a model before it stabilizes, then it is possible to insert an extra (identical) state into this sequence, without disturbing the truth of ψ. Since this enlarged model is smaller (with respect to $\preceq^g$) than the original, the original model could not have been a minimal model of ψ. The length of such a sequence depends on the depth of nesting of temporal operators in ψ. We will now formalize these ideas.

Definition 9.11 (Depth) The *depth of nesting* of temporal operators in a formula φ, denoted $\mathrm{depth}(\varphi)$, is defined inductively as follows:

- $\mathrm{depth}(\varphi) = 0$, if $\varphi \in \mathcal{L}_{S5}$;
- $\mathrm{depth}(\alpha \wedge \beta) = \max\{\mathrm{depth}(\alpha), \mathrm{depth}(\beta)\}$;
- $\mathrm{depth}(\neg\alpha) = \mathrm{depth}(\alpha)$;
- $\mathrm{depth}(P\alpha) = \mathrm{depth}(F\alpha) = \mathrm{depth}(\alpha) + 1$.

The first lemma states that in a sequence of identical states, formulae with small enough depth cannot discriminate between states in the middle of the sequence. The Lemmas 9.12 and 9.13 and Observation 9.14 are also valid for non-subjective formulae.

Lemma 9.12 If $\mathcal{M}$ is a TEL-model such that for some $N \geq 1$, $s \geq N$:

$$\mathcal{M}_s = \mathcal{M}_{s+i} = \mathcal{M}_{s-i} \quad \text{for all } 1 \leq i \leq N,$$

then for all φ with $\text{depth}(\varphi) < N$ and $1 \leq j \leq N - \text{depth}(\varphi)$:

$$(\mathcal{M}, s-j) \models \varphi \Leftrightarrow (\mathcal{M}, s) \models \varphi \Leftrightarrow (\mathcal{M}, s+j) \models \varphi.$$

Proof: By induction on φ, where the only interesting cases are the temporal operators (the abbreviation 'i.h.' stands for induction hypothesis):

$F\alpha$: Let $1 \leq j \leq N - \text{depth}(F\alpha)$. The implications from right to left are trivial, so we will only prove $(\mathcal{M}, s-j) \models F\alpha \Rightarrow (\mathcal{M}, s+j) \models F\alpha$.

Suppose $(\mathcal{M}, s-j) \models F\alpha$. There exists $n \in \mathbb{N}$, $n > s-j$ with $(\mathcal{M}, n) \models \alpha$. If $n > s+j$ then $(\mathcal{M}, s+j) \models F\alpha$, so suppose $s - j < n \leq s + j$. Then there are a number of cases:

- If $n = s - k$ with $1 \leq k < j$ then $1 \leq k < j \leq N - \text{depth}(F\alpha) < N - \text{depth}(\alpha)$ and with the i.h. we get $(\mathcal{M}, s) \models \alpha$.
- If $n = s$ then $(\mathcal{M}, s) \models \alpha$.
- If $n = s + k$ with $1 \leq k \leq j$ then $1 \leq k \leq j \leq N - \text{depth}(F\alpha) < N - \text{depth}(\alpha)$, so by the i.h.we get $(\mathcal{M}, s) \models \alpha$.

So we have $(\mathcal{M}, s) \models \alpha$ and $1 \leq j+1 \leq N - (\text{depth}(F\alpha) - 1) = N - \text{depth}(\alpha)$, so by the i.h. we have $(\mathcal{M}, s + (j+1)) \models \alpha$ so $(\mathcal{M}, s+j) \models F\alpha$.

$P\alpha$: Analogous to $F\alpha$.

$\square$

We will often use this lemma with $j = 1$ and $N = \text{depth}(\varphi) + 1$.

The example model $\mathcal{M}$ depicted in Figure 9.1 shows that we really need that many identical states. In this model, nothing is known at time point 0, and p is known from time point 1 onwards, and q is known from time point 5. We have $(\mathcal{M}, 3) \not\models G(Kq)$ but $(\mathcal{M}, 3+1) \models G(Kq)$ (we need an extra Kp state between 4 and 5); also $(\mathcal{M}, 2-1) \models H(\neg Kp)$ but $(\mathcal{M}, 2) \not\models H(\neg Kp)$ (we need an extra Kp state between 0 and 1).

The next lemma shows that if we have a sequence of identical states, a middle state can be duplicated or removed without changing the truth of formulae with sufficiently small depth of operator-nesting:

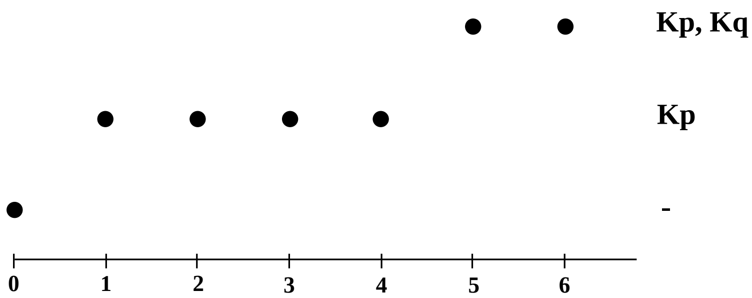

Figure 9.1: The model $\mathcal{M}$.

Lemma 9.13 Let $\mathcal{M}$ be a model as in Lemma 9.12. Define $f : \mathbb{N} \to \mathbb{N}$ as follows:

$$f(n) = \begin{cases} n & \text{if } n \leq s \\ n-1 & \text{if } n > s \end{cases}$$

and let $\mathcal{N}$ be the model defined by $\mathcal{N}_i = \mathcal{M}_{f(i)}$ for all $i \in \mathbb{N}$. Then for all formulae φ with $\text{depth}(\varphi) \leq N$ we have:

$$(\mathcal{N}, i) \models \varphi \Leftrightarrow (\mathcal{M}, f(i)) \models \varphi \qquad \text{for all } i \in \mathbb{N}.$$

Proof: By induction on φ, where the only non-trivial cases are the operators (for which we will take H and G):

$H\varphi$: Suppose $(\mathcal{N}, i) \models H\varphi$. Take $k < f(i)$. Then there exists $t < i$ such that $f(t) = k$ and then $(\mathcal{N}, t) \models \varphi$, so by the i.h. $(\mathcal{M}, k) \models \varphi$. Thus $(\mathcal{M}, f(i)) \models H\varphi$. Suppose $(\mathcal{M}, f(i)) \models H\varphi$.

- If $i \leq s$: take $k < i$ then $f(k) < f(i)$, so $(\mathcal{M}, f(k)) \models \varphi$ and by the i.h. $(\mathcal{N}, k) \models \varphi$. We have $(\mathcal{N}, i) \models H\varphi$.
- If $i \geq s+1$: take $k < i$;
  - If $k \neq s$ then $f(k) < f(i)$, so $(\mathcal{M}, f(k)) \models \varphi$ and by the i.h. $(\mathcal{N}, k) \models \varphi$.
  - If $k = s$ then $s-1 < f(i)$, so $(\mathcal{M}, s-1) \models \varphi$. As $\text{depth}(H\varphi) \leq N$ we have $1 \leq 1 \leq N - \text{depth}(\varphi)$ and by Lemma 9.12 we have $(\mathcal{M}, s) \models \varphi$ and by the i.h. $(\mathcal{N}, s) \models \varphi$, or $(\mathcal{N}, k) \models \varphi$.

So we have $(\mathcal{N}, i) \models H\varphi$.

$G\varphi$: Analogous.

□

Figure 9.2 sketches the situation with $N = 2$.

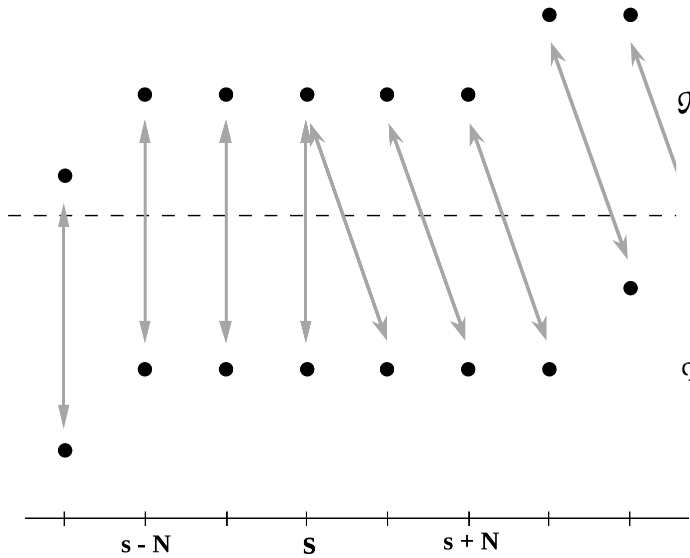

Figure 9.2: Duplicating or removing a state.

Another way of proving this lemma is to show that there exist bisimulations up to N between these two models. The main use of the lemma lies in the possibility of enlarging or reducing sequences of identical states in a model without disturbing truth of formulae with sufficiently small depth of nesting.

Observation 9.14 For the models $\mathcal{M}$,$\mathcal{N}$ of Lemma 9.13 the following holds: if $\mathcal{M}$ is conservative then $\mathcal{N}$ is conservative and vice versa, $\mathcal{N} \preceq^g \mathcal{M}$, and if there exists $t \geq s + N$ such that $\mathcal{M}_t \prec \mathcal{M}_{t+1}$ then $\mathcal{N} \prec^g \mathcal{M}$.

Proof: Take $s \in \mathbb{N}$, then $\mathcal{N}_s = \mathcal{M}_{f(s)}$. Since $f(s) \leq s$ and $\mathcal{M}$ is conservative we have $\mathcal{M}_{f(s)} \preceq \mathcal{M}_s$ so $\mathcal{N}_s \preceq \mathcal{M}_s$. If there exists $t \geq s + N$ such that $\mathcal{M}_t \prec \mathcal{M}_{t+1}$ then $\mathcal{N}_{t+1} = \mathcal{M}_{f(t+1)} = \mathcal{M}_t \prec \mathcal{M}_{t+1}$. □

This observation and the previous lemma allow us to conclude that for each formula there is a time point such that the minimal models of the formula must stabilize before this point. From now on we will again restrict ourselves to subjective formulae.

Lemma 9.15 Suppose the propositional signature P consists of n atoms. If a conservative model $\mathcal{M}$ of signature P is a $\preceq^g$- minimal model of a subjective formula φ then it stabilizes on or before time point $(2^n - 1) \cdot 2 \cdot \text{depth}(\varphi)$.

Proof: First we will show that a minimal model $\mathcal{M}$ of φ cannot have more than $2 \cdot \text{depth}(\varphi)$ successive identical states before it stabilizes. Suppose $\mathcal{M} \models_{\preceq^g} \varphi$ and it has at least $2 \cdot \text{depth}(\varphi) + 1$ successive identical states before it stabilizes. So there exists $s \geq \text{depth}(\varphi)$ such that $\mathcal{M}_s = \mathcal{M}_{s+i} = \mathcal{M}_{s-i}$ for all $1 \leq i \leq \text{depth}(\varphi)$, and $t \geq s + \text{depth}(\varphi)$ such that $\mathcal{M}_t \prec \mathcal{M}_{t+1}$. Now consider the model $\mathcal{N}$ as described in Lemma 9.13. Since $\mathcal{M} \models \varphi$ we have $\mathcal{N} \models \varphi$, and by Observation 9.14 we have $\mathcal{N} \prec^g \mathcal{M}$. Therefore $\mathcal{M}$ cannot be a minimal model of φ.

As P has n atoms, there exist 2^n different propositional models. Since a conservative model $\mathcal{M}$ consists of a decreasing sequence of (non-empty) sets of propositional models, there are at most $2^n - 1$ points s such that $\mathcal{M}_s \prec \mathcal{M}_{s+1}$. If $\mathcal{M}$ is a $\preceq^g$-minimal model of φ then there can be at most $2 \cdot \text{depth}(\varphi)$ successive identical states before it stabilizes, and therefore $\mathcal{M}$ must stabilize on or before time point $(2^n - 1) \cdot 2 \cdot \text{depth}(\varphi)$. □

Lemma 9.16 Suppose P is finite. For a conservative model $\mathcal{M}$, $s \in \mathbb{N}$ and a formula φ it is decidable whether $(\mathcal{M}, s) \models \varphi$.

Proof: Suppose we have a conservative model $\mathcal{M}$ and $s \in \mathbb{N}$. By Observation 9.10, $\mathcal{M}$ stabilizes at some point s_0. It is easily seen from Lemma 9.12 that for a formula φ we have $(\mathcal{M}, t) \models \varphi \Leftrightarrow (\mathcal{M}, u) \models \varphi$ for all $t, u \geq s_0 + \text{depth}(\varphi)$. Then use induction on φ. □

Most importantly, it is decidable if a model is a minimal model of a subjective formula:

Lemma 9.17 Let P be finite. For a conservative model $\mathcal{M}$ and a subjective formula φ it is decidable whether $\mathcal{M} \models_{\preceq^g} \varphi$.

Proof: First, we need to check whether $\mathcal{M} \models \varphi$, which is equivalent to checking $(\mathcal{M}, 0) \models \Box\varphi$, which is decidable by Lemma 9.16. Suppose P has n atoms. If $\mathcal{M}$ stabilizes after time point $(2^n - 1) \cdot 2 \cdot \text{depth}(\varphi)$ it is not a minimal model of φ by Lemma 9.15. So suppose $\mathcal{M} \models \varphi$ and $\mathcal{M}$ stabilizes on or before time point $(2^n - 1) \cdot 2 \cdot \text{depth}(\varphi)$.

In order to check whether $\mathcal{M} \models_{\preceq^g} \varphi$ we have to see if there exists a conservative model smaller (in the sense of $\preceq^g$) than $\mathcal{M}$ which satisfies φ. Of course in general there are an infinite number of conservative models smaller than $\mathcal{M}$, but we will show that we only have to consider models which stabilize not later than time point $(2^n - 1) \cdot (4 \cdot \text{depth}(\varphi) + 1)$. In other words, we will show that if there exists a conservative model smaller than $\mathcal{M}$ satisfying φ, there also exists such a model which stabilizes on or before point $(2^n - 1) \cdot (4 \cdot \text{depth}(\varphi) + 1)$. The converse of this statement is of course trivial.

Suppose we have a conservative model $\mathcal{N}$ with $\mathcal{N} \prec^g \mathcal{M}$ and $\mathcal{N} \models \varphi$, and let s be the stabilizing point of $\mathcal{N}$. If $s \leq (2^n - 1) \cdot (4 \cdot \text{depth}(\varphi) + 1)$ then we are done, so suppose not. Now consider the following procedure for constructing a model $\mathcal{N}'$: if there exists a sequence of more than $2 \cdot \text{depth}(\varphi) + 1$ successive identical states in $\mathcal{N}$ between time points $(2^n - 1) \cdot 2 \cdot \text{depth}(\varphi)$ and s then we delete as many points from this sequence until it has length $2 \cdot \text{depth}(\varphi) + 1$. Lemma 9.13 ensures that we can do this without disturbing the truth of φ. It is also easy to see that the result is conservative and still (strictly) smaller than $\mathcal{M}$. Let $\mathcal{N}'$ be the model which results from applying this procedure for every such sequence. Then $\mathcal{N}' \models \varphi$ and $\mathcal{N}' \prec^g \mathcal{M}$. Let s' be the stabilizing point of $\mathcal{N}'$. Then in $\mathcal{N}'$ there are at most $2^n - 1$ points t with $(2^n - 1) \cdot 2 \cdot \text{depth}(\varphi) \leq t < s$ and $\mathcal{N}'_t \prec \mathcal{N}'_{t+1}$. Between such points there are at most $2 \cdot \text{depth}(\varphi) + 1$ identical states and therefore $s \leq (2^n - 1) \cdot 2 \cdot \text{depth}(\varphi) + (2^n - 1) \cdot (2 \cdot \text{depth}(\varphi) + 1) = (2^n - 1) \cdot (4 \cdot \text{depth}(\varphi) + 1)$.

It is easy to see that, given the finite signature, there are only a finite number of conservative models which stabilize not later than time point $(2^n - 1) \cdot (4 \cdot \text{depth}(\varphi) + 1)$. For each such model $\mathcal{N}$ we can check whether $\mathcal{N} \prec^g \mathcal{M}$ (only the first $(2^n - 1) \cdot (4 \cdot \text{depth}(\varphi) + 1)$ time points have to be considered), and we can check if $\mathcal{N} \models \varphi$ (again decidable). If we find such a model then $\mathcal{M} \not\models_{\preceq^g} \varphi$, otherwise $\mathcal{M} \models_{\preceq^g} \varphi$. $\square$

Now we are ready to prove decidability of entailment in MTEL:

Theorem 9.18 (Decidability of entailment in MTEL) For two subjective formulae φ, ψ it is decidable whether $\varphi \models_{\preceq^g} \psi$.

Proof: We can take the signature P to consist of the atoms occurring in φ and ψ. Suppose there are n such atoms. Then Lemma 9.15 states that we only have to consider models which stabilize not later than time point $(2^n - 1) \cdot 2 \cdot \text{depth}(\varphi)$, and since the signature is finite, there are only finitely many such models. For each such model $\mathcal{M}$ it is decidable by Lemma 9.17 whether $\mathcal{M} \models_{\preceq^g} \varphi$. Now we only have to check for each of these (finitely many) $\preceq^g$-minimal models $\mathcal{M}$ of φ whether $\mathcal{M} \models \psi$, which is decidable by Lemma 9.16. $\square$

Of course the procedure given in the proof is quite inefficient.

Having established that both TELC and MTEL are decidable, in the next section we will look at the complexity of these notions, and in particular whether the minimization process has a structural impact on complexity.

9.1.3 Complexity

We will first give a brief overview of the relevant concepts of complexity theory needed in the rest of this subsection. This is meant as a reminder for the reader, not as an introduction to this field (see [Joh90] for a good introduction). Especially the Polynomial Hierarchy (PH) will concern us here. The Polynomial Hierarchy is a hierarchy of classes of problems of increasing complexity. The two most well-known complexity classes in PH are P and NP. The basic notion in defining complexity classes is the Turing Machine (TM). The class P consists of all problems solvable by a deterministic TM running in time polynomial in the length of the input. Problems solvable by a non-deterministic TM running in polynomial time form the class NP. For any complexity class C, the class co-C consists of the problems whose complement is in C. In order to define the other classes in PH, we need the notion of an oracle TM, which is a TM that has access to an oracle for a particular decision problem: all instances of that problem can be solved in one time step by consulting the oracle. Formally, if C is a complexity class then the class NP^{C} consists of those problems solvable by a nondeterministic TM with access to an oracle for a problem in C, running in time polynomial in the input size. Now set:

$$\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}, \quad \text{and for } k \geq 0:$$
$$\Sigma_{\mathrm{k}+1}^{\mathrm{P}} = \mathrm{NP}^{\Sigma_{\mathrm{k}}^{\mathrm{P}}} \text{ and } \Pi_{k+1}^{\mathrm{P}} = \text{co-}\Sigma_{\mathrm{k}+1}^{\mathrm{P}}.$$

Note that $\Sigma_1^{\mathrm{P}} = \mathrm{NP}$ and $\Pi_1^{\mathrm{P}} = \text{co-NP}$. For a problem p, if for any problem in class C there is a polynomial transformation of that problem to p, then p is called C-hard. If p is in C and is C-hard, it is called C-complete. If a C-hard problem can be (polynomially) transformed to p, then p is also C-hard.

In order to study the complexity we will first look at satisfiability of TELC. Without loss of generality we restrict ourselves to satisfiability of subjective formulae in time point 0. For future use we give the following definition. (Here O^i stands for a sequence of O operators of length i, where $O \in \{P, H, F, G, \Box\}$. Furthermore $O^0\alpha$ stands for α.)

Definition 9.19 For $i \in \mathbb{N}$ define $at_i := P^i\top \wedge H^{i+1}\bot$.

It is easy to see that $(\mathcal{M}, j) \models at_i$ if and only if $i = j$.

Definition 9.20 (TELC(0)-SAT) A subjective formula φ is in TELC(0)-SAT if there exists a TELC-model $\mathcal{M}$ such that $(\mathcal{M}, 0) \models \varphi$.

Remark 9.21 It is easy to see that TELC(0)-SAT is polynomially reducible (and vice versa) to satisfiability (in any time point): φ is satisfiable if and only if $\varphi \vee F\varphi$ is in TELC(0)-SAT , and φ is in TELC(0)-SAT if and only if $\Box(at_0 \to \varphi)$ is satisfiable.

Definition 9.22 (Size of a TELC-model) For a TELC-model $\mathcal{M}$ we call its stabilizing point the *size* of $\mathcal{M}$, denoted size($\mathcal{M}$). If $\mathcal{M}$ does not stabilize then size($\mathcal{M}$) $= \infty$.

Definition 9.23 (Subformula) Let Subf(φ) denote the subformulae of φ, where maximal S5-subformulae of φ are not further decomposed, and let SubfS5(φ) denote the set of subformulae of φ which are in $\mathcal{L}_{S5}$.

We give an example to clarify this definition: $\mathrm{Subf}(G(Kp \wedge Kq)) = \{G(Kp \wedge Kq), Kp \wedge Kq\}$ and $\mathrm{SubfS5}(G(Kp \wedge Kq)) = \{Kp \wedge Kq, Kp, Kq, p, q\}$. So $\mathrm{Subf}(\varphi) \cup \mathrm{SubfS5}(\varphi)$ is the set of all subformulae of φ.

First we will prove a small-model theorem for TELC. Let $\mathrm{length}(\varphi)$ denote the length of the formula φ as a string.

Lemma 9.24 (Small model theorem)
If a subjective formula φ is in TELC(0)-SAT then there exists a TELC-model $\mathcal{M}$ such that $(\mathcal{M}, 0) \models \varphi$, $\mathrm{size}(\mathcal{M}) \leq 4 \cdot (\mathrm{length}(\varphi))^2$, and for all $i \in \mathbb{N}$ the S5-model $\mathcal{M}_i$ contains not more than $2 \cdot \mathrm{length}(\varphi)$ valuations.

Proof: We may take the signature P finite (in fact, we can take it to be the set of atoms occurring in φ). Let $\mathcal{L}_0$ denote the propositional language based on P. Suppose for some TELC-model $\mathcal{N}$ we have $(\mathcal{N}, 0) \models \varphi$ and let $s_{\mathcal{N}}$ be the stabilizing point of $\mathcal{N}$.

Now let $A = \{\psi, \neg\psi \mid \psi \in \mathrm{SubfS5}(\varphi) \cap \mathcal{L}_0\}$ and for $i \in \mathbb{N}$: $B(i) = \{K\psi \mid \psi \in A, \mathcal{N}_i \models K\psi\} \cup \{\neg K\psi \mid \psi \in A, \mathcal{N}_i \not\models K\psi\}$. Based on these sets we will define a TELC-model $\mathcal{N}'$:

- For each $\neg K\psi \in B(s_{\mathcal{N}})$ choose a valuation $m \in \mathrm{Val}(P)$ such that $m \not\models \psi$ and $m \models \alpha$ for each $K\alpha \in B(s_{\mathcal{N}})$ (such a valuation exists since $(\mathcal{N}, s_{\mathcal{N}}) \not\models K\psi$ and $(\mathcal{N}, s_{\mathcal{N}}) \models K\alpha$ for each $K\alpha \in B(s_{\mathcal{N}})$). Let M be the set of these valuations. We have $M \models B(s_{\mathcal{N}})$. If there are no formulae $\neg K\psi \in B(s_{\mathcal{N}})$ then choose any valuation m with $m \models \alpha$ for each $K\alpha \in B(s_{\mathcal{N}})$ (which again exists). Set $\mathcal{N}'_j = M$ for all $j \geq s_{\mathcal{N}}$. It easy to verify that $\mathcal{N}'_j \models B(j)$ for all $j \geq s_{\mathcal{N}}$.

- Now using induction on $s_{\mathcal{N}} > j \geq 0$:
  Let $B(j) \setminus B(j+1) = \{\neg K\psi_1, \ldots, \neg K\psi_n\}$ (because $\mathcal{N}$ is conservative there will be no formulae $K\psi$ in this set!). For $k = 1 \ldots n$ choose a valuation m_k with $m_k \not\models \psi_k$ and $m \models \alpha$ for each $K\alpha \in B(j)$ (again such valuations exist). Let $\mathcal{N}'_j = \mathcal{N}'_{j+1} \cup \{m_1, \ldots, m_k\}$. It is again easy to verify that $\mathcal{N}'_j \models B(j)$.

The resulting model $\mathcal{N}'$ has the following properties:

1. $\mathcal{N}'$ is a TELC-model.

2. $\mathcal{N}'_j \models B(j)$ for all $j \in \mathbb{N}$.

3. The number of valuations of $\mathcal{N}'_j$ is smaller than the number of elements in A ($\leq 2 \cdot \mathrm{length}(\varphi)$).

4. $(\mathcal{N}', 0) \models \varphi$: take $\psi \in \mathrm{Subf}(\varphi) \cap \mathcal{L}_{S5}$ (which must be subjective!). Then using a normal form described in [MH95] it is easy to see that ψ is equivalent to a formula $\psi' = \delta_1 \vee \ldots \vee \delta_m$ with for $i = 1 \ldots m$: $\delta_i = K\varphi_{1,i} \wedge \ldots \wedge K\varphi_{k(i),i} \wedge \neg K\psi_{1,i} \wedge \ldots \wedge \neg K\psi_{l(i),i}$ with $\varphi_{j,k}, \psi_{j,k} \in A$. So using the second property we

have:

$$\begin{array}{lcll} \mathcal{N}'_i \models K\varphi_{j,k} & \Leftrightarrow & \mathcal{N}_i \models K\varphi_{j,k} & \text{and} \\ \mathcal{N}'_i \models \neg K\psi_{j,k} & \Leftrightarrow & \mathcal{N}_i \models \neg K\psi_{j,k} & \text{so} \end{array}$$

$\mathcal{N}'_i \models \psi' \Leftrightarrow \mathcal{N}_i \models \psi'$ so $\mathcal{N}'_i \models \psi \Leftrightarrow \mathcal{N}_i \models \psi$. An easy induction gives: for all $i \in \mathbb{N}$, for all $\psi \in \text{Subf}(\varphi)$: $(\mathcal{N}', i) \models \psi \Leftrightarrow (\mathcal{N}, i) \models \psi$ and therefore $(\mathcal{N}', 0) \models \varphi$.

5. The number of i for which $\mathcal{N}'_i \prec \mathcal{N}'_{i+1}$ is less than $2 \cdot \text{length}(\varphi)$: real changes occur at most once for each $\neg K\psi$ with $\psi \in A$ and A contains at most $2 \cdot \text{length}(\varphi)$ elements.

Now construct the model $\mathcal{M}$ as follows: for each sequence of more than $2{\cdot}\text{depth}(\varphi){+}1$ identical states in $\mathcal{N}'$, before its stabilizing point, delete (as many) states from this sequence until it has length $2 \cdot \text{depth}(\varphi) + 1$. Let $\mathcal{M}$ be the resulting model. Now Lemma 9.13 ensures that $(\mathcal{M}, 0) \models \varphi$. Furthermore $2 \cdot \text{depth}(\varphi) + 1 \leq 2 \cdot \text{length}(\varphi)$ so that $\text{size}(\mathcal{M}) \leq (2 \cdot \text{length}(\varphi))^2$. □

With this lemma we can show that TELC(0)-SAT is in NP, using methods similar to those in e.g. [SC85], [Lad77]:

Theorem 9.25 TELC(0)-SAT is in NP.

Proof: For a subjective formula φ we present the following nondeterministic algorithm to verify if φ is in TELC(0)-SAT. A nondeterministic Turing Machine M guesses $4 \cdot (\text{length}(\varphi))^2$ Kripke models $\mathcal{M}_i$ with each not more than $2 \cdot \text{length}(\varphi)$ valuations, such that $\mathcal{M}_i \supseteq \mathcal{M}_{i+1}$. Call this model $\mathcal{M}$, remaining constant after time point $4 \cdot (\text{length}(\varphi))^2$. Then it verifies if $(\mathcal{M}, 0) \models \varphi$ as follows: for each $i \in \{0, ..., 4 \cdot (\text{length}(\varphi))^2 + \text{length}(\varphi)\}$, M maintains a set $label(i)$ which is initialized to the empty set and at the end will contain the subformulae of φ true at time point i. Now for each $\psi \in \text{Subf}(\varphi)$ we do the following (starting with the S5- subformulae, and treating ψ only if all of its subformulae have already been treated): for each $i \in \{0, ..., 4 \cdot (\text{length}(\varphi))^2 + \text{length}(\varphi)\}$ update $label(i)$ as follows:

1. Add $\psi \in \mathcal{L}_{S5}$ to $label(i)$ iff $\mathcal{M}_i \models \psi$ (this can be checked in time polynomial in number of states in $\mathcal{M}_i$, using a labeling algorithm similar to the one described here, see e.g. [HM85a]).

2. Add $\neg\psi$ to $label(i)$ iff $\psi \notin label(i)$.

3. Add $\alpha \wedge \beta$ to $label(i)$ iff $\alpha \in label(i)$ and $\beta \in label(i)$.

4. Add $F\alpha$ to $label(i)$ iff $\alpha \in label(j)$ for some $j > i$ (If $i = 4 \cdot (\text{length}(\varphi))^2 + \text{length}(\varphi)$ then add $F\alpha$ to $label(i)$ iff $\alpha \in label(i)$).

5. Add $P\alpha$ to $label(i)$ iff $\alpha \in label(j)$ for some $j < i$.

Now we have $(\mathcal{M}, 0) \models \varphi$ if and only if $\varphi \in \mathit{label}(0)$ at the end of this procedure. It is easy to verify that this algorithm works properly in time polynomial in length(φ). Lemma 9.24 ensures that there is a guess for which M halts in an accepting state if and only if φ is in TELC(0)-SAT. $\square$

This gives us:

Corollary 9.26 TELC satisfiability is NP-complete.

Proof: The reduction given in Remark 9.21 ensures that TELC satisfiability is in NP, and clearly a propositional formula φ is satisfiable if and only if $M\varphi$ is TELC-satisfiable, and as satisfiability of propositional formulae is NP-complete, TELC satisfiability is also NP-complete. $\square$

We would like to show that the minimization of models makes the consequence relation more complex, and we can do this using the reduction of Ground S5 to minimal conservative consequence, which will be described later (see Proposition 9.40).

Proposition 9.27 Entailment of MTEL ($\models_{\preceq^g}$) is Π_3^{P}-hard.

Proof: The reduction of Proposition 9.40 is clearly polynomial, and Ground S5 is Π_3^{P}-complete [DNR97]. $\square$

So entailment in MTEL is harder than TELC-consequence (which is Π_1^{P}-complete, or co-NP-complete), provided that the polynomial hierarchy does not collapse (see [Joh90]).

In Chapters 6 and 7, execution and expressiveness of a sublanguage of $\mathcal{L}_{TEL}$, consisting of theories of reasoning, is studied. We will now look at the complexity of entailment in MTEL restricted to this language. As entailment is a relation between formulae, we should consider finite theories of reasoning, and take the conjunction of its elements. Furthermore, we also allow formulae that describe facts, of the form $K\alpha$ with α propositional (let $\mathcal{L}_0$ be the propositional language). The conclusion formula may be arbitrary (but subjective, of course).

Definition 9.28 The language $\mathcal{L}'$ is the smallest set such that:

1. If $\alpha \in \mathcal{L}_0$ then $K\alpha \in \mathcal{L}'$,
2. If $\alpha, \beta, \gamma, \psi$ and $\varphi \in \mathcal{L}_0$ then $H_0(K\alpha) \wedge H_0(\neg K\beta) \wedge K\gamma \wedge \neg F(K\psi) \rightarrow X(K\varphi) \in \mathcal{L}'$, and
3. If $\varphi, \psi \in \mathcal{L}'$ then $\varphi \wedge \psi \in \mathcal{L}'$.

For $\varphi \in \mathcal{L}'$ and ψ any subjective TEL-formula, we define $\varphi \models^{\mathrm{ttr}}_{\preceq^g} \psi$ if and only if $\varphi \models_{\preceq^g} \psi$.

Since we can reduce sceptical consequence of default logic to this fragment (see Theorem 5.5) and sceptical consequence is Π_2^{P}-complete (see [Got92], [Sti92], see also [PS92]), $\models^{\mathrm{ttr}}_{\preceq^g}$ is Π_2^{P}-hard. However, it is no harder than that:

Proposition 9.29 $\models^{\mathrm{ttr}}_{\preceq^g}$ is Π_2^{P}-complete.

Proof: We will describe a nondeterministic Turing Machine M with access to an NP-oracle for determining whether *not* $\varphi \models^{\mathrm{ttr}}_{\preceq^g} \psi$ (similar to the proofs in [Sti92], [PS92] or [Got92]). A minimal model of φ can have no identical states before it stabilizes. For each conjunct $H_0(K\alpha) \wedge H_0(\neg K\beta) \wedge K\gamma \wedge \neg F(K\delta) \rightarrow X(K\epsilon)$ in φ , M guesses a time point $i \geq 1$ but less than the number n of these conjuncts, from which time onwards ϵ will be assumed to hold (or it guesses that ϵ will never hold). Denote for $i \in \{0, \ldots, n\}$, the set of formulae assumed to hold at i plus the formulae α for which there is a conjunct $K\alpha$ in φ, by $A(i)$. Then M uses the NP- oracle to perform the following:

1. Let $f(\epsilon)$ be the point from which ϵ is assumed to hold (so $f(\epsilon) \in \{1, \ldots, n, \infty\}$). Now it checks for all $i \in \{1, \ldots, n\}$ if $\{K\epsilon \mid f(\epsilon) \leq i\} \cup \{\neg K\epsilon \mid f(\epsilon) > i\}$ is S5- satisfiable (using the oracle; note that S5-satisfiability is in NP). If not, it halts in a rejecting state (the guess does not induce a TELC-model).

2. For each conjunct $H_0(K\alpha) \wedge H_0(\neg K\beta) \wedge K\gamma \wedge \neg F(K\delta) \rightarrow X(K\epsilon)$ and for each time point $i \in \{0, \ldots, n\}$ it computes whether $A(0) \models \alpha$, whether $A(0) \not\models \beta$, whether $A(i) \models \gamma$ and whether for no $i < j \leq n$, $A(j) \models \delta$, using the NP-oracle. If this is true for no time point then it checks whether ϵ is assumed never to hold; otherwise it takes the first such point and checks whether ϵ is assumed to hold from the next time point on. If these conditions are violated then M halts in a rejecting state (the guess does not induce a minimal model of φ).

3. It checks if the induced TELC-model satisfies ψ using a labeling algorithm similar to the one in the proof of Theorem 9.25, and using the NP-oracle for checking whether $A(i) \models K\alpha$ for the base case. If this is the case then in this minimal model of φ, ψ holds, so M halts in a rejecting state (the guess does not induce a minimal model of φ in which ψ fails). Otherwise it halts in an accepting state (the guess induces a minimal model of φ in which ψ does not hold).

This nondeterministic algorithm is polynomial in φ and ψ (using an NP-oracle for propositional consequence and S5-satisfiability) so the converse of $\models^{\mathrm{ttr}}_{\preceq^g}$ is in Σ_2^{P} which implies that $\models^{\mathrm{ttr}}_{\preceq^g}$ is in Π_2^{P}. Together with Π_2^{P}-hardness this gives the desired result. $\square$

Apart from default logic, sceptical consequence relations of many other well-known nonmonotonic logics such as McDermott and Doyle's nonmonotonic logic, autoepistemic logic and nonmonotonic logic N are Π_2^{P}-complete ([Got92]) which means that we can reduce these relations to MTEL entailment (or even $\models^{\mathrm{ttr}}_{\preceq^g}$), using a polynomial reduction (autoepistemic logic was translated into MTEL$^{\overline{*}}$ earlier). Further research is needed to find these reductions.

We would also like to have an upper bound on the complexity of MTEL entailment. In order to get this, we need to sharpen some previous lemmas. Lemma 9.15 gave an upper bound on the size of minimal models of φ, but it is not polynomial in the length of φ. We already know that the length of a sequence of identical states in a minimal model is polynomially bounded, so we will try to find a polynomial bound on the number of transitions between non-identical states in a minimal model. The key is that in a minimal model of φ, after such a transition occurs, the agent will know (at least) one of the subformulae of φ he did not know before. In fact, a minimal model of φ is uniquely determined by the subformulae of φ which are true at any moment in time. We will now make this formal.

Definition 9.30 For a subjective formula φ, define $A(\varphi) = \{\psi, \neg\psi \mid \psi \in \mathcal{L}_0 \cap \mathrm{SubfS5}(\varphi)\}$. A TELC-model $\mathcal{M}$ of φ is called *based on* φ (abbreviated bo(φ)) if there exist sets $A(i)$ for each $i \in \mathbb{N}$ with $A(0) \subseteq A(1) \subseteq \ldots \subseteq A(\varphi)$ and $\mathcal{M}_i = \mathit{Mod}(A(i)) = \{m \in \mathrm{Val}(P) \mid m \models A(i)\}$.

Lemma 9.31 Suppose P is finite. If $\mathcal{M} \models_{\preceq^g} \varphi$ then $\mathcal{M}$ is bo(φ) and $\mathrm{size}(\mathcal{M}) \leq 4 \cdot (\mathrm{length}(\varphi))^2$.

Proof: Suppose $\mathcal{M}$ is not based on φ. Define $A(i) = \{\alpha \mid \alpha \in A(\varphi) \text{ and } \mathcal{M}_i \models K\alpha\}$ and let $\mathcal{N}_i = \mathit{Mod}(A(i))$. Clearly $A(0) \subseteq A(1) \subseteq \ldots \subseteq A(\varphi)$ so $\mathcal{N}$ is a TELC-model and $\mathcal{N} \prec^g \mathcal{M}$. Furthermore, for all $\alpha \in \mathcal{L}_0 \cap \mathrm{SubfS5}(\varphi)$ we have $\mathcal{M}_i \models K\alpha \Leftrightarrow \mathcal{N}_i \models K\alpha$ and $\mathcal{M}_i \models M\alpha \Leftrightarrow \mathcal{N}_i \models M\alpha$, so using the same argument as in the proof of Lemma 9.24 we have $\mathcal{N} \models \varphi$. This contradicts the assumption that $\mathcal{M} \models_{\preceq^g} \varphi$, so $\mathcal{M}$ is based on φ. But then the number of changes in $\mathcal{M}$ (the points $i \in \mathbb{N}$ where $\mathcal{M}_i \prec \mathcal{M}_{i+1}$) cannot be larger than the number of elements of $A(\varphi)$ and in-between such updates there cannot be sequences of identical states longer than $2 \cdot \mathrm{depth}(\varphi) + 1$ so $\mathrm{size}(\mathcal{M}) \leq 4 \cdot (\mathrm{length}(\varphi))^2$. □

Notice that a model $\mathcal{M}$ based on φ can equivalently be described by giving for each formula in $A(\varphi)$ the time point at which it is known in $\mathcal{M}$, or 'infinity' if this is never the case. We have a similar result for models which refute that $\mathcal{M}$ is a minimal model of φ:

Lemma 9.32 If $\mathcal{M} \models \varphi$ but $\mathcal{M} \not\models_{\preceq^g} \varphi$, then there exists a TELC-model $\mathcal{N}$ such that $\mathcal{N} \prec^g \mathcal{M}$, $\mathcal{N} \models \varphi$ and $\mathcal{N}$ is based on φ with $\mathrm{size}(\mathcal{N}) \leq \mathrm{size}(\mathcal{M}) + 4 \cdot (\mathrm{length}(\varphi))^2$.

Proof: Suppose $\mathcal{M} \models \varphi$ but $\mathcal{M} \not\models_{\preceq^g} \varphi$ then there is a TELC-model $\mathcal{M}'$ with $\mathcal{M}' \prec^g \mathcal{M}$ and $\mathcal{M}' \models \varphi$. In the same way as in the proof of Lemma 9.31 we can make a model $\mathcal{M}''$ which is a model of φ based on φ and $\mathcal{M}'' \preceq^g \mathcal{M}'$. Now from any sequence of identical states in $\mathcal{M}''$ after size($\mathcal{M}$) but before size($\mathcal{M}''$) with length more than $2 \cdot \text{depth}(\varphi) + 1$ we can delete states until it has length $2 \cdot \text{depth}(\varphi) + 1$. Let $\mathcal{N}$ be the resulting model (this construction is the same as the one used in the proof of Lemma 9.17). So we have $\mathcal{N} \prec^g \mathcal{M}$, $\mathcal{N} \models \varphi$ and $\mathcal{N}$ is based on φ. Furthermore, $\mathcal{N}$ has less than $2 \cdot \text{length}(\varphi)$ updates, and sequences between size($\mathcal{M}$) and size($\mathcal{N}$) have length no greater than $2 \cdot \text{depth}(\varphi) + 1$, so $\text{size}(\mathcal{N}) \leq \text{size}(\mathcal{M}) + 2 \cdot \text{length}(\varphi) \cdot 2 \cdot \text{length}(\varphi) = \text{size}(\mathcal{M}) + 4 \cdot (\text{length}(\varphi))^2$. □

Lemma 9.33 Deciding for a formula φ and a model $\mathcal{M}$ based on φ whether $\mathcal{M} \models_{\preceq^g} \varphi$ is in Π_2^{P}.

Proof: We assume the model $\mathcal{M}$ encoded as described in the remark after Lemma 9.31: there is a function $f : A(\varphi) \to \mathbb{N} \cup \{\infty\}$ such that $f(\alpha)$ gives the time point from which α is known. We will show that deciding whether $\mathcal{M} \not\models_{\preceq^g} \varphi$ is in Σ_2^{P} by describing a nondeterministic Turing Machine M with access to an NP-oracle. Let $\text{size}(\mathcal{M}) = \max(f[A(\varphi)] \setminus \{\infty\})$ (if $f[A(\varphi)] = \{\infty\}$, then let $\text{size}(\mathcal{M}) = 0$). First we check if $\text{size}(\mathcal{M}) \leq 4 \cdot (\text{length}(\varphi))^2$; if not we halt in an accepting state. Otherwise we use a labeling algorithm as described earlier to check if $\mathcal{M} \models \varphi$. The range of time points we have to check is from 0 to $\text{size}(\mathcal{M}) + \text{length}(\varphi)$. The subformulae in $\text{Subf}(\varphi) \cap \mathcal{L}_{S5}$ are treated as follows: for such a formula α and time point i it is checked (using the NP- oracle) if $\{K\epsilon \mid f(\epsilon) \leq i\} \cup \{\neg K\epsilon \mid f(\epsilon) > i\} \models_{S5} \alpha$. If so, α is added to $\mathit{label}(i)$, otherwise not. If $\mathcal{M} \not\models \varphi$, M halts in an accepting state (certainly $\mathcal{M} \not\models_{\preceq^g} \varphi$). Otherwise M guesses a TELC- model $\mathcal{N}$ by guessing a function $g : A(\varphi) \to \mathbb{N} \cup \{\infty\}$ such that:

1. $f(\epsilon) \leq g(\epsilon)$,
2. either $g(\epsilon) \leq \text{size}(\mathcal{M}) + 4 \cdot (\text{length}(\varphi))^2$ or $g(\epsilon) = \infty$, and
3. for at least one $\epsilon \in A(\varphi)$ we have $g(\epsilon) > f(\epsilon)$.

Then it checks for $i \in \{0, \ldots, \text{size}(\mathcal{M}) + 4 \cdot (\text{length}(\varphi))^2\}$ whether $\{K\epsilon \mid g(\epsilon) \leq i\} \cup \{\neg K\epsilon \mid g(\epsilon) > i\}$ is S5-consistent, using the oracle. If not, we halt in a rejecting state (g does not describe a TELC-model). Otherwise we know that g induces a TELC-model $\mathcal{N}$ with $\mathcal{N} \prec^g \mathcal{M}$ (if such a guess is not possible then we halt in a rejecting state because $\mathcal{M} \models_{\preceq^g} \varphi$). Next we use the labeling algorithm to check if $\mathcal{N} \models \varphi$; if not we halt in a rejecting state, otherwise in an accepting state: $\mathcal{N}$ is a smaller model of φ. It is clear that the algorithm works in polynomial time (using the NP-oracle).

Lemma 9.32 ensures that there is a guess for which M halts in an accepting state if and only if $\mathcal{M} \not\models_{\preceq^g} \varphi$. Thus deciding if $\mathcal{M} \not\models_{\preceq^g} \varphi$ is in Σ_2^{P} so the complement is in Π_2^{P}. □

Theorem 9.34 Deciding whether $\varphi \models_{\preceq^g} \psi$ is in Π_3^{P}.

Proof: We will show that deciding whether *not* $\varphi \models_{\preceq^g} \psi$ is in Σ_3^{P} by giving a nondeterministic Turing Machine M with access to an oracle for a problem in Π_2^{P}. First M guesses a TELC-model $\mathcal{M}$ based on φ by guessing a function $f : A(\varphi) \to \mathbb{N} \cup \{\infty\}$ such that for all $\epsilon \in A(\varphi)$ either $f(\epsilon) \leq 4 \cdot (\text{length}(\varphi))^2$ or $f(\epsilon) = \infty$. Then it checks for $i \in \{0, \ldots, 4 \cdot (\text{length}(\varphi))^2\}$ whether $\{K\epsilon \mid f(\epsilon) \leq i\} \cup \{\neg K\epsilon \mid f(\epsilon) > i\}$ is S5-consistent, using the oracle. If not it halts in a rejecting state (f does not induce a TELC-model). Now it uses the Π_2^{P}-oracle to determine if $\mathcal{M} \models_{\preceq^g} \varphi$. If not it halts in a rejecting state. Otherwise it uses a labeling algorithm to check if $\mathcal{M} \models \psi$ (as in the proof of the previous lemma, using the Π_2^{P}-oracle for S5-consequence); if this is true M halts in a rejecting state, otherwise in an accepting state. The algorithm works in polynomial time, and Lemma 9.31 ensures there is a guess for which M halts in an accepting state if and only if *not* $\varphi \models_{\preceq^g} \psi$. So as this is in Σ_3^{P}, the complement is in Π_3^{P}. $\square$

Combining this with Proposition 9.27, we immediately get:

Corollary 9.35 Entailment in MTEL ($\models_{\preceq^g}$) is Π_3^{P}-complete.

9.1.4 Conclusions and related work

In this section, we gave Hilbert-style axiomatizations for the logics TEL and TELC, and showed MTEL to be decidable. Furthermore, the computational complexity of these logics was investigated. The minimization process of MTEL really makes the consequence relation more complex: TELC is co-NP-complete, whereas MTEL is Π_3^{P}-complete. If we restrict the antecedent to theories of reasoning, the consequence relation of MTEL is Π_2^{P}-complete (and Π_2^{P} is the complexity class where consequence relations of many nonmonotonic formalisms, such as default logic and auto-epistemic logic, reside).

The fact that the interaction between the epistemic part and the temporal part is very limited in TELC (only conservativity provides such a link) allowed us to use techniques from [FG92]. It probably also accounts for the fact that TELC is no harder than its constituent logics (S5 is co-NP-complete, and so is the linear temporal logic with only F and P), an effect also noted in [HV89]. This latter publication lists a number of complexity results for logics of knowledge and time (varying the number of agents, the language — both temporal and epistemic operators — and some other parameters).

In further research, the effect of extensions of TELC and MTEL on the complexity can be investigated. Such extensions include allowing more agents, dropping conservativity, adding temporal operators such as Next or Until, taking another epistemic logic (such as S4), adding observations and communications, etc. Furthermore, we would like to find a deduction system for MTEL, possibly in the same way as was done for Ground S5 ([EV98]).

9.2 Monotonicity and persistence in preferential logics

The logic MTEL we have studied before, is easily seen to be nonmonotonic: it holds that $Kp \models_{\preceq^g} \neg Kq$ but $Kp \wedge Kq \not\models_{\preceq^g} \neg Kq$. This nonmonotonicity is a general feature of so-called *preferential* logics, of which MTEL belongs: it consists of a class of models (conservative TEL-models) and a preference relation ($\preceq^g$) between them. The current section explores the relationship between preferential logics and monotonicity restricted to subsets of the language.

9.2.1 Restricted monotonicity

Over the past decades, many non-classical logics for Artificial Intelligence have been defined and investigated. The need for such logics arose from the unsuitability of classical logics to describe defeasible reasoning. These classical logics are monotonic, which means that their consequence relation ($\mid\!\sim$) satisfies:

$$\forall \alpha, \beta, \varphi : (\alpha \mid\!\sim \beta \Rightarrow \alpha \wedge \varphi \mid\!\sim \beta) \qquad (\textit{Monotonicity})$$

This means that whenever we learn new information (φ) and add this to what we already know (α), all the old theorems (β) are still derivable. This is clearly undesirable when describing defeasible reasoning. Therefore, monotonicity is not satisfied by many logics for Artificial Intelligence.

On the other hand, monotonicity is a very attractive feature from a practical point of view. When learning new information, we do not have to start all over again, but we can retain our old conclusions, and focus on deriving possible new ones. Furthermore, when we have a lot of information, we are allowed to focus on only part of it. Conclusions derived from this part are then automatically also valid when considering all the information we have (this is sometimes called *local reasoning*).

Even though it is clear that we do not want monotonicity to hold in general for logics for defeasible reasoning, it might be worthwhile to investigate restricted variants of monotonicity. In the past, such variants have been defined which allow us to keep the old theorems, when either the new information follows from the old premise (this variant is called Cautious Monotonicity in [KLM90], see also Section 9.3) or its negation can not be derived from the old premise (this is called Rational Monotonicity in [KLM90]).

We will take a somewhat different perspective, and consider two classes of formulae: the class of formulae that can always be added to a premise without invalidating old conclusions (we say these formulae *respect monotonicity*), and the class of formulae which can always be retained as conclusions, no matter which new information is added to the premise (we say these formulae are *conservative*). The advantages of monotonicity sketched above would still hold when we restrict φ to the class of formulae that respect monotonicity, or when we restrict β to be conservative.

Whether such classes exist, and what these classes are, depends of course on the particular nonmonotonic logic considered. We will focus here on an important class of nonmonotonic logics: the class of *preferential logics* [Sho87, Sho88]. These logics are based on a monotonic logic (such as propositional logic, predicate logic or modal logic) augmented with a preference order on its models. The nonmonotonic consequences of a formula α are those formulae which are true in all models of α which are minimal in the preference order among all models of α, analogously to the definition of MTEL (an extensive discussion of preferential logics is provided in [Ben89]). We will give a formal definition.

Definition 9.36 (Preferential logic) A preferential logic consists of a language $\mathcal{L}$, a class of models Mod together with a satisfaction relation $\models$ between models and formulae, and a partial order $\preceq$ on Mod. A model $m \in \mathsf{Mod}$ is called a *minimal model* of a formula α (denoted $m \models_{\preceq} \alpha$) if $m \models \alpha$ and for all models n, if $n \preceq m$ and $n \models \alpha$ then $n = m$. Preferential entailment ($\models_{\preceq}$) between formulae is defined as follows for $\alpha, \beta \in \mathcal{L} : \alpha \models_{\preceq} \beta$ if β is true in all minimal models of α.

Our presentation uses a partial order, i.e., a reflexive, antisymmetric and transitive relation, as the preferential logics defined in Section 4.4 have these properties. Shoham [Sho87] uses a strict partial order, i.e., an irreflexive transitive relation, with a slightly different notion of minimal model. The presentations can be translated into each other.

It will turn out that formulae whose truth is preserved when going to more preferred or less preferred models, play an important role with respect to the two classes of formulae defined above (the class of formulae that respect monotonicity, and the class of conservative formulae). We will first give a definition.

Definition 9.37 (Persistence) Given a preferential logic $(\mathcal{L}, \mathsf{Mod}, \models, \preceq)$, a formula $\alpha \in \mathcal{L}$ is called *downward persistent* in this logic, if

$$\forall m, n \in \mathsf{Mod} : (m \models \alpha \text{ and } n \preceq m) \Rightarrow n \models \alpha,$$

and it is called *upward persistent* if

$$\forall m, n \in \mathsf{Mod} : (n \models \alpha \text{ and } n \preceq m) \Rightarrow m \models \alpha.$$

In the next subsection, we will introduce some preferential logics to illustrate the material in the rest of this section.

9.2.2 Some preferential logics

The logic MTEL is easily seen to be a preferential logic in the sense of Definition 9.36. In this subsection, we will describe two more preferential logics: Ground S5 and circumscription. Since we have already defined preferential entailment in general, for each logic we only have to give its ingredients, i.e., $\mathcal{L}, \mathsf{Mod}, \models$, and $\preceq$. The preferential entailment relation is then fixed by Definition 9.36. We will first consider Ground S5.

Ground S5

Ground S5 is a nonmonotonic modal logic for autoepistemic reasoning, originally proposed by Halpern and Moses [HM85b]. Their aim was to formalize statements of the form "I *only* know φ". It allows, for example, to derive that an agent which only knows p, does not know q. Ground S5 falls into the general scheme of ground nonmonotonic modal logics [DNR97]. A lot of interest is devoted to logics of minimal knowledge, see for example [Lev90, ST94, Che97, Hal97].

Semantically, states in which an agent only knows φ, are states in which φ is known, but otherwise the amount of knowledge is minimal. We will use a modal propositional language to express the knowledge of the agent, and S5 will be the monotonic logic. We will give a treatment of Ground S5 slightly different, but equivalent to the one given in [HM85b].

The language of Ground S5 consists of the subjective formulae of S5, and the models are the normal S5-models described in Subsection 4.1.1.

A subjective formula describes the knowledge of an agent, but we want to formalize that this is *all* the agent knows. Therefore we are looking for models in which the knowledge of the agent is minimal, or in other words, in which the ignorance of the agent is maximal. But this is formalized exactly by the ordering $\preceq$ on IS^{ep} introduced in Definition 2.5. To summarize, we will give a formal definition of Ground S5.

Definition 9.38 (Ground S5) Ground S5 is the preferential logic with the subjective formulae of $\mathcal{L}_{\mathrm{S5}}$ as its language, the set of normal S5-models as its class of models, the satisfaction relation $\models_{S5}$ and the ordering $\preceq$ of Definition 2.5. We will denote preferential entailment (as defined in Definition 9.36) of Ground S5 by $\models^{\mathrm{GS5}}$.

The reader can now check that, for instance, $Kp \models^{\mathrm{GS5}} \neg Kq$. The (unique) minimal S5-model of Kp consists of all propositional valuations in which p is true, and this indeed contains a model in which q is false. The entailment relation is nonmonotonic since $Kp \wedge Kq \not\models^{\mathrm{GS5}} \neg Kq$. Another example illustrates the minimality of the agent's knowledge: $Kp \vee Kq \models^{\mathrm{GS5}} \neg(Kp \wedge Kq)$.

Let us define a consequence relation $\mid\!\sim$ by $\varphi \mid\!\sim \psi$ if $K\varphi \models^{\mathrm{GS5}} K\psi$. Then it turns out that this is the consequence relation of [HM85b], apart from the fact that Halpern and Moses only defined it for premises which have a unique minimal model. Premises with a unique minimal model are called *honest*. To give an example, the formula Kp is honest, but $Kp \vee Kq$ is not: both the S5-model consisting of all valuations in which p is true, and the model with all valuations in which q is true, are minimal models.

Our logic MTEL can now be seen as a 'temporalization' of Ground S5: the knowledge (or beliefs) of the agent are minimized; in Ground S5 only the knowledge at one fixed instance in time is minimized, whereas in MTEL all knowledge over time is minimized. Ground S5 indeed subsumes MTEL in the sense that we can (trivially) embed Ground S5 in MTEL. Before showing this more formally, we will first slightly alter the definition of MTEL. The reason is that we will later make the general assumption on preferential logics that the language contains negation, and that $m \models \neg\varphi$ if and only if $m \not\models \varphi$ (Assumption 9.42). This is not the case for the notion of satisfaction of a formula in a model, $\mathcal{M} \models \varphi$, used in the definition of MTEL. Therefore we will use $(\mathcal{M}, 0) \models \varphi$ for satisfaction in a model.

Definition 9.39 (Anchored MTEL) In this section we define satisfaction of a formula φ in a TELC-model $\mathcal{M}$, which we will denote as $\mathcal{M} \models \varphi$, by $(\mathcal{M}, 0) \models \varphi$. Let *TCIS* denote the set of TELC-models. In this section, MTEL is the preferential logic which uses the subjective TEL-formulae as its language, *TCIS* as the class of models, satisfaction of a formula in a model as above, and $\preceq^g$ as its ordering.

We could call this variant MTEL$'$, but in order to keep the notation clean, we will refer to the variant used in this section again simply as MTEL. The two definitions can be translated into each other, since $(\mathcal{M}, 0) \models \varphi$ if and only if $(\mathcal{M}, t) \models \neg P\top \to \varphi$ for all $t \in \mathbb{N}$ and $(\mathcal{M}, t) \models \varphi$ for all $t \in \mathbb{N}$ if and only if $(\mathcal{M}, 0) \models \varphi \wedge G\varphi$. The assumption leading to the altered definition of MTEL is also one of the reasons we gave a slightly different presentation of Ground S5 (using subjective formulae).

We can now prove the following.

Proposition 9.40 (Ground S5 in MTEL) Let φ and ψ be subjective S5-formulae. Then

$$\varphi \models^{\mathrm{GS5}} \psi \Leftrightarrow \varphi \models_{\preceq^g} \psi.$$

Proof: If M is an S5-model which is a minimal model of φ in Ground S5, then the model $\mathcal{M}$ defined by $\mathcal{M}_s = M$ for all $s \in \mathbb{N}$ can easily be seen to be a $\preceq^g$-minimal model of φ. On the other hand, any $\preceq^g$-minimal model $\mathcal{M}$ of φ must be constant, and $\mathcal{M}_0$ is then a minimal model of φ in Ground S5. From these two observations, which are not hard to prove, the proposition follows in a straightforward manner. □

Circumscription

One of the earliest approaches to nonmonotonic reasoning is circumscription [McC77, McC80, Dav80a, Lif94, Eth88], a preferential logic based on first-order predicate logic. The main idea behind circumscription is a kind of completeness of information given to us: "the premises as stated give us 'the whole truth' about the matter" [Ben89]. This leads to at least two kinds of minimality: predicate-minimality and domain-minimality. The intuition behind predicate-minimality is that for some relevant property (predicate), all objects that have this property are explicitly said to have this property in the premise. This allows us to formulate defaults stating that all normal objects have some property. Minimizing abnormality will allow us to conclude an object has this property, unless we can deduce from the premise that this object is abnormal. The intuition behind domain-minimality, is that the domain (of discourse) contains no other objects than those that can be deduced to exist from the premise. (This intuition is strongly tied to the domain-closure assumption of [Rei80a].) These two kinds of minimality are formalized by two variants of circumscription. Both of them will be treated below.

The classical logic underlying circumscription is first-order predicate logic. We assume a standard first-order language $\mathcal{L}$ with a finite number of predicate symbols, including equality. We will also assume that the language contains no function or constant symbols. This is not a severe limitation, since we can eliminate function and constant symbols by introducing new predicate symbols (see [Dav80a]). We will first give the definition of the orderings and then define predicate and domain circumscriptive consequence.

Definition 9.41

1. Let P be a predicate symbol in the language $\mathcal{L}$. For a structure M for the language, P^M denotes the interpretation of P in M (so P^M is a subset of $dom(M)^n$, where $dom(M)$ is the domain of M, and n is the arity of P). For two structures M, N, we say M is *P-preferred* to N, denoted $M \leq_P N$, if they have the same domain, the same interpretation of predicate symbols other than P, and $P^M \subseteq P^N$. Predicate circumscription of P is the preferential logic which uses first-order predicate logic for the language, models and satisfaction relation, augmented with the ordering $\leq_P$. We will denote preferential entailment (as defined in Definition 9.36) in this logic by $\models_P^{\text{PC}}$.

2. For two structures M, N for the language $\mathcal{L}$, we say N is a *substructure* of M, denoted $N \leq_d M$, if the domain of N is a subset of the domain of M, and the interpretation of each predicate symbol in N is the restriction of the corresponding interpretation in M to $dom(N)$. Domain circumscription is the preferential logic which uses first-order predicate logic for the language, models and satisfaction relation, augmented with the ordering $\leq_d$. We will denote preferential entailment in this logic by $\models^{\text{DC}}$.

3. If we restrict the model class to finite structures, the resulting preferential logics are called *finite predicate circumscription* and *finite domain circumscription*.

We refer the reader to the references given above for standard results and motivation of circumscription.

9.2.3 Respecting monotonicity

In this subsection we will study formulae which respect monotonicity. We will first make some basic assumptions about the (underlying logic of the) preferential logic.

Assumption 9.42 From now on we will assume that any preferential logic satisfies the following:

- the language has conjunction ($\wedge$) and $m \models \varphi \wedge \psi \Leftrightarrow m \models \varphi$ and $m \models \psi$.
- the language has implication ($\rightarrow$) and $m \models \varphi \rightarrow \psi \Leftrightarrow m \not\models \varphi$ or $m \models \psi$.
- the language has negation ($\neg$) and $m \models \neg\varphi \Leftrightarrow m \not\models \varphi$.

Note that Ground S5, MTEL (the anchored version) and both versions of circumscription satisfy these assumptions (for Ground S5 it is essential that the language only contains subjective formulae).

We will now give a formal definition of respecting monotonicity.

Definition 9.43 (Respecting monotonicity) Given a preferential logic, we say a formula φ *respects monotonicity*, if

$$\forall \alpha, \beta : \alpha \models_{\preceq} \beta \Rightarrow \alpha \wedge \varphi \models_{\preceq} \beta.$$

We can immediately identify a class of formulae that respect monotonicity:

Proposition 9.44 Downward persistent formulae respect monotonicity.

Proof: Suppose φ is downward persistent. Let α, β be formulae and suppose $\alpha \models_{\preceq} \beta$. Let m be a minimal model of $\alpha \wedge \varphi$. Then it is also a minimal model of α. For suppose it is not, then there exists $n \preceq m$, $n \neq m$ and $n \models \alpha$. Since $m \models \varphi$ and φ is downward persistent, we have $n \models \varphi$. But then $n \models \alpha \wedge \varphi$ which contradicts the assumption that m was a minimal model of $\alpha \wedge \varphi$. Since m is a minimal model of α and $\alpha \models_{\preceq} \beta$, we have $m \models \beta$. We have proved that $\alpha \wedge \varphi \models_{\preceq} \beta$. Thus, φ respects monotonicity. $\square$

Of course, both valid and unsatisfiable sentences are downward persistent. But the question is whether non-trivial downward persistent formulae exist. For the preferential logics introduced in Subsection 9.2.2, the answer is affirmative.

Definition 9.45 (DIAM) Define the class of S5-formulae *DIAM* by:

$$DIAM ::= M(\varphi) \mid DIAM \wedge DIAM \mid DIAM \vee DIAM \mid M(DIAM)$$

where φ is propositional.

Formulae from *DIAM* essentially only contain the M operator (the 'diamond' of S5, and not the 'box' operator K). Formulae in this class are the *only* subjective formulae (up to equivalence) which are downward persistent in Ground S5.

Theorem 9.46 A subjective S5-formula φ is downward persistent in Ground S5 if and only if it is S5-equivalent to a formula in *DIAM*.

Proof: Using Proposition 4.3 it is easy to see that a formula equivalent to one in *DIAM* is downward persistent in Ground S5. So suppose φ is downward persistent. Then we may restrict the signature P to be finite. If there is no S5-model M such that $M \models_{S5} \varphi$, then φ is S5-equivalent to the formula $M(p \wedge \neg p) \in \mathit{DIAM}$. So suppose such a model M exists. Then let $A = \min\{N \subseteq \mathrm{Val}(P) \mid N \neq \emptyset \text{ and } N \models_{S5} \varphi\}$, where for a set $\mathcal{B}$ of S5-models, $\min \mathcal{B} = \{N \in \mathcal{B} \mid \text{there is no } M \in \mathcal{B} \text{ such that } M \text{ is a proper subset of } N\}$. For $m \in \mathrm{Val}(P)$, define $\alpha_m := \bigwedge\{p \mid p \in P, m \models p\} \wedge \bigwedge\{\neg p \mid p \in P, m \not\models p\}$ and for an S5-model N, $\varphi_N = \bigwedge\{M\alpha_m \mid m \in N\}$. It is easy to see that for an S5-model M we have that $N \subseteq M$ if and only if $M \models_{S5} \varphi_N$. Finally, let

$$\psi = \bigvee\{\varphi_N \mid N \in A\}.$$

Then ψ is in *DIAM*, and ψ is equivalent to φ:

- Suppose that $N \models_{S5} \varphi$. Then there exists an $M \in A$ with $M \subseteq N$, so $N \models_{S5} \varphi_M$ and therefore $N \models_{S5} \psi$.
- Suppose that $N \models_{S5} \psi$. Then there must be an $M \in A$ such that $N \models_{S5} \varphi_M$, so that $M \subseteq N$ and $M \models_{S5} \varphi$. But as φ is downward persistent, this means that $N \models_{S5} \varphi$.

We have proved that $N \models_{S5} \psi \Leftrightarrow N \models_{S5} \varphi$, and as both formulae are subjective, this implies that they are S5-equivalent. □

So in Ground S5 there is a non-empty class of downward persistent formulae, that respect monotonicity by Proposition 9.44. Essentially, these formulae only say something about the ignorance of the agent.

One might think that formulae from *DIAM* are completely uninteresting, and never yield any new insights in Ground S5. The converse of monotonicity for these formulae, $\alpha \wedge \varphi \models^{\mathrm{GS5}} \beta \Rightarrow \alpha \models^{\mathrm{GS5}} \beta$, however, does not hold, even when φ is consistent with α. We do not have that $Kp \vee Kq \models^{\mathrm{GS5}} Kq$, whereas we do have that $(Kp \vee Kq) \wedge M(\neg p) \models^{\mathrm{GS5}} Kq$ with $M(\neg p) \in \mathit{DIAM}$. So knowledge of ignorance can be useful.

An analogous result holds for minimal temporal epistemic logic.

Definition 9.47 (TD)

1. Define

$$\mathit{TD} ::= \mathit{DIAM} \mid \mathit{TD} \wedge \mathit{TD} \mid \mathit{TD} \vee \mathit{TD} \mid F(\mathit{TD}) \mid G(\mathit{TD}) \mid P(\mathit{TD}) \mid H(\mathit{TD}).$$

2. For two subjective TEL-formulae φ, ψ:

$$\varphi \sim_0 \psi \Leftrightarrow_{def} \text{ for all TELC-models } \mathcal{M} : (\mathcal{M}, 0) \models \varphi \Leftrightarrow (\mathcal{M}, 0) \models \psi.$$

TD stands for 'temporal diamond' formulae. As was the case for Ground S5, in MTEL too *TD* contains all downward persistent formulae, up to equivalence.

Theorem 9.48 In MTEL, a formula φ is downward persistent if and only if it is equivalent (in the sense of $\sim_0$) to a formula in *TD*.

Proof: For a subjective (!) formula φ in *TD* one can easily prove that for all TELC-models $\mathcal{M}, \mathcal{N}$ and $i \in \mathbb{N}$: if $\mathcal{M} \preceq^g \mathcal{N}$ and $(\mathcal{N}, i) \models \varphi$ then $(\mathcal{M}, i) \models \varphi$. This implies that a formula equivalent (in the sense of $\sim_0$) to one in *TD* is downward persistent.

Suppose φ is a subjective downward persistent formula. We will construct its equivalent in *TD*. If there is no TELC-model $\mathcal{M}$ such that $(\mathcal{M}, 0) \models \varphi$ then φ is $\sim_0$-equivalent to $\bot$. Note that $\bot$ is equivalent to $M(p \wedge \neg p)$ which is a subjective formula in *TD*. We may again restrict the signature P to the atoms occurring in φ. Suppose the propositional signature P has r atoms. For a set of TELC-models $\mathfrak{B}$ define $\max \mathfrak{B} = \{\mathcal{M} \in \mathfrak{B} \mid \text{there is no } \mathcal{N} \in \mathfrak{B} \text{ with } \mathcal{M} \prec^g \mathcal{N}\}$. If there is a TELC-model $\mathcal{M}$ such that $(\mathcal{M}, 0) \models \varphi$, then we define $\mathfrak{A} = \max\{\mathcal{M} \mid (\mathcal{M}, 0) \models \varphi\}$. Suppose $(\mathcal{M}, 0) \models \varphi$ and $\mathcal{M}$ stabilizes after time point $(2^r - 1) \cdot (2 \cdot \text{depth}(\varphi) + 1)$. Then we can delete points in sequences of more than $(2 \cdot \text{depth}(\varphi) + 1)$ identical states before the stabilizing point, without disturbing the truth of φ. If we do this for each such a sequence we end up with a model of φ which is larger (with respect to $\preceq^g$) than $\mathcal{M}$ and stabilizes not later than $(2^r - 1) \cdot (2 \cdot \text{depth}(\varphi) + 1)$. Thus: $\mathfrak{A} = \max\{\mathcal{M} \mid (\mathcal{M}, 0) \models \varphi \text{ and } \mathcal{M} \text{ stabilizes not later than } (2^r - 1) \cdot (2 \cdot \text{depth}(\varphi) + 1)\}$. As the set we take the maximal elements of, is non-empty and finite and the relation $\prec^g$ on TELC-models is transitive and irreflexive, $\mathfrak{A}$ is non-empty and finite. Note that the argument used here (for maximal models) is similar to the one used for minimal models in the proof of Lemma 9.15: there the idea was that a model which is too long can be enlarged (yielding a smaller model with respect to $\preceq^g$), whereas here the idea is that if a model is too long, it can be reduced (yielding a bigger model with respect to $\preceq^g$).

Suppose $\text{Val}(P) = \{m_1, \ldots, m_n\}$ (with of course $n = 2^r$). Again define for $j = 1 \ldots n$: $\alpha_j := \bigwedge\{p \mid p \in P, m_j \models p\} \wedge \bigwedge\{\neg p \mid p \in P, m_j \not\models p\}$. Now define for $i = 1 \ldots n$ and for a TELC-model $\mathcal{M}$:

$$n(i, \mathcal{M}) = \sup\{j \in \mathbb{N} \mid m_i \in \mathcal{M}_j\} \quad \text{where } \sup \emptyset = -\infty.$$

Let

$$\psi(i, \mathcal{M}) = \begin{cases} \Box(at_{n(i,\mathcal{M})} \to M\alpha_i) & \text{if } n(i, \mathcal{M}) \in \mathbb{N} \\ \Box(M\alpha_i) & \text{if } n(i, \mathcal{M}) = \infty \\ \top & \text{if } n(i, \mathcal{M}) = -\infty \end{cases}$$

(Note that $\top$ is equivalent to $M(p \vee \neg p)$). Furthermore, define $\psi_{\mathcal{M}} = \bigwedge\{\psi(i, \mathcal{M}) \mid i = 1 \ldots n\}$. Now it can easily be proven that $(\mathcal{N}, 0) \models \psi_{\mathcal{M}} \Leftrightarrow \mathcal{N} \preceq^g \mathcal{M}$: the formulae $\psi(i, \mathcal{M})$ make sure that the valuation m_i is in $\mathcal{N}_t$ at least until the last time point s for which m_i is in $\mathcal{M}_s$. Finally, define:

$$\psi = \bigvee\{\psi_{\mathcal{M}} \mid \mathcal{M} \in \mathfrak{A}\}.$$

Then ψ is in *TD* and $\varphi \sim_0 \psi$:

- Suppose $(\mathcal{M}, 0) \models \varphi$. Then there exists $\mathcal{N} \in \mathfrak{A}$ with $\mathcal{M} \preceq^g \mathcal{N}$ (!), so $(\mathcal{M}, 0) \models \psi_{\mathcal{N}}$ and $(\mathcal{M}, 0) \models \psi$.
- Suppose $(\mathcal{M}, 0) \models \psi$. Then there exists $\mathcal{N} \in \mathfrak{A}$ with $(\mathcal{M}, 0) \models \psi_{\mathcal{N}}$, so $\mathcal{M} \preceq^g \mathcal{N}$ and as $\mathcal{N} \in \mathfrak{A}$ we have $(\mathcal{N}, 0) \models \varphi$, and φ was downward persistent, so $(\mathcal{M}, 0) \models \varphi$.

□

As in the case of Ground S5, these formulae express (temporal) ignorance of the agent.

Definition 9.49 (Positive and universal formulae) A first-order predicate formula is *negative in* a predicate P, if all occurrences of the predicate P are in the scope of an odd number of negations. A formula is *universal* if it is of the form $\forall x_1 \ldots x_n \psi$ where ψ is quantifier free.

The following result links these formulae to downward persistence in circumscription. The first is a variant of Lyndon's theorem and is folklore (we leave the details to the reader); the second result is known as the Łoś-Tarski theorem (Theorem 3.2.2 in [CK90]).

Theorem 9.50

1. A first-order predicate formula φ is downward persistent in predicate circumscription (of P) if and only if it is equivalent to a formula that is negative in P.
2. A first-order predicate formula φ is downward persistent in domain circumscription if and only if it is equivalent to a universal formula.

So downward persistent formulae in predicate circumscription essentially only say something about elements not having property P (besides the other properties they mention), and downward persistent formulae in domain circumscription essentially only mention universal properties (and do not say anything about the existence of objects).

For our examples, we have shown that non-trivial classes of formulae that respect monotonicity exist. The question is whether there are more such formulae, besides those that are downward persistent. We will give a criterion that ensures that there are no more formulae that respect monotonicity.

Definition 9.51 (Expressibility of preference) A preferential logic satisfies *expressibility of preference* if the following holds:

$$\forall m \in \mathsf{Mod} : \exists \varphi^m \in \mathcal{L} : \forall n \in \mathsf{Mod} : (n \models \varphi^m \Leftrightarrow m \preceq n).$$

The formula φ^m expresses: "I am less preferred than m," and describes exactly those models which are larger in the preferential ordering. The criterion of expressibility of preference poses a requirement on the expressiveness of the language, given its semantics. We will prove that in preferential logics that satisfy the condition in this definition, the downward persistent formulae are the only ones that respect monotonicity. The above condition can be generalized by taking into account equivalent models; we have not done this immediately as it makes things rather cumbersome. If whenever $n \preceq m$ and $m \equiv k$ (where $m \equiv k$ means that m and k satisfy the same formulae), there exists a model l such that $l \equiv n$ and $l \preceq k$, then we can generalize the condition to: $\forall m \in \mathsf{Mod} : \exists \varphi^m \in \mathcal{L} : \forall n \in \mathsf{Mod} : (n \models \varphi^m \Leftrightarrow \exists k \in \mathsf{Mod} : m \equiv k \;\&\; k \preceq n)$.

Theorem 9.52 (Only if ...) For a preferential logic that satisfies expressibility of preference we have: if a formula respect monotonicity, then it is downward persistent.

Proof: Suppose a formula φ is not downward persistent, then there exist models m and n such that $m \models \varphi$, $n \not\models \varphi$ and $n \preceq m$. Define $\alpha = \varphi^n \wedge (\varphi \to \varphi^m)$ and $\beta = \neg\varphi$. First we claim that $\alpha \models_{\preceq} \beta$. Since $n \preceq n$, we have $n \models \varphi^n$, and as $n \not\models \varphi$ we get $n \models \alpha$. Furthermore, for any model k, if $k \models \alpha$ then in particular $k \models \varphi^n$ so $n \preceq k$. Therefore, n is the only minimal model of α, and since $n \not\models \varphi$, we have $n \models \beta$. On the other hand, $\alpha \wedge \varphi \not\models_{\preceq} \beta$: $n \preceq m$ so $m \models \varphi^n$ and $m \preceq m$ so $m \models \varphi^m$ from which we conclude that $m \models \alpha$ so $m \models \alpha \wedge \varphi$. Furthermore, for any model k, if $k \models \alpha \wedge \varphi$, then $k \models \varphi$ and $k \models \varphi \to \varphi^m$ so $k \models \varphi^m$. From this it follows that $m \preceq k$, but this means that m is a (actually, the only) minimal model of $\alpha \wedge \varphi$ and $m \models \varphi$ so $m \not\models \beta$. We conclude that φ does not respect monotonicity, since we have found formulae α and β such that $\alpha \models_{\preceq} \beta$ but $\alpha \wedge \varphi \not\models_{\preceq} \beta$. $\square$

It may seem that the condition of expressibility of preference is too restrictive. However, we will see that it is useful for the examples.

Proposition 9.53 For Ground S5, MTEL and finite predicate and domain circumscription, only downward persistent formulae respect monotonicity.

Proof: Remark that all of these logics satisfy Assumption 9.42. First consider Ground S5. Let us first take the language to be finite (that is, P is finite). Take any S5-model M. For each propositional valuation m, define the formula α^m by $\alpha^m = \bigwedge\{p \in P \mid m \models p\} \wedge \bigwedge\{\neg p \mid p \in P, m \not\models p\}$. This is a well-defined formula since P is finite. Now construct $\varphi^M = \bigwedge\{K(\neg\alpha^m) \mid m \notin M\}$, which is again a well-defined formula since $\mathrm{Val}(P)$ is finite. It can easily be seen that any S5-model N satisfies φ^M if and only if $M \preceq N$. So expressibility of preference is satisfied, whence Theorem 9.52 ensures that only downward persistent formulae respect monotonicity for this finite language. Now let P be arbitrary, and suppose φ in this language respects monotonicity. Then it is easy to see that if we restrict the language to atoms occurring in φ, it still respects monotonicity, so it is downward persistent in the restricted language. It follows easily that φ is also downward persistent in the full language. Note the difference with the formulae φ_N defined in the proof of Theorem 9.46 which express "I am *more* preferred than N". The same holds for the formulae $\varphi^{\mathcal{M}}$ below in relation to the formulae $\psi_{\mathcal{M}}$ of the proof of Theorem 9.48.

For MTEL, the same considerations make it sufficient to give a formula $\varphi^{\mathcal{M}}$ for a finite language only, so let us take P finite. Let $\mathcal{M}$ be a TELC-model. Then every S5-model $\mathcal{M}(i)$ is a finite set of propositional valuations. Since the sequence $\{\mathcal{M}(i)\}$ is decreasing with respect to set-inclusion (as $\mathcal{M}$ is conservative), there will be an index k such that $\mathcal{M}(j) = \mathcal{M}(k)$ for all $j > k$. Now define:

$$\varphi^{\mathcal{M}} = \bigwedge\{\Box(at_i \to \varphi^{\mathcal{M}(i)}) \mid 0 \leq i \leq k\},$$

where $\varphi^{\mathcal{M}(i)}$ is the formula as defined in the case of Ground S5 for the S5-model $\mathcal{M}(i)$. It is easy to show that $\mathcal{N} \models \varphi^{\mathcal{M}}$ if and only if $\mathcal{M} \preceq \mathcal{N}$.

For finite circumscription, we need the more general definition of expressibility of preference hinted at before (in first-order logic, there may be equivalent models: different models that satisfy the same first-order formulae). Here we need not restrict the language. In predicate circumscription, the required formula φ^M for a finite structure M expresses: (i) the exact number of elements of the domain of M, (ii) for which of these elements P holds, and (iii) for all other predicates Q it expresses for which elements Q holds, and for which its negation holds. In domain circumscription, the required formula φ^M for a finite structure M expresses the fact that there are (at least) as many elements as in M, and for each predicate Q, it expresses for which of these elements Q holds, and for which elements its negation holds. □

It is not possible to find the required formula φ^M in (non-finite) circumscription in general: for infinite structures we are not in general able to express the number of elements, and we can not describe the entire extensions of predicates in general. Indeed, the above result does not hold for domain circumscription. It is still an open question whether it holds for predicate circumscription.

Proposition 9.54 For domain circumscription, there exists a first-order predicate formula which respects monotonicity but is not downward persistent.

Proof: Consider the first-order language $\mathcal{L} = \{<, =\}$, and let φ be a sentence stating that $<$ is a dense linear ordering without begin- or endpoint. This is a complete theory (see Theorem 4 in [Rab77]), which means that for any $\alpha \in \mathcal{L}$, either $\varphi \models \alpha$ or $\varphi \models \neg\alpha$. Now suppose $\alpha \models^{\text{DC}} \beta$. If $\varphi \models \neg\alpha$ then $\alpha \wedge \varphi$ is inconsistent, so $\alpha \wedge \varphi \models^{\text{DC}} \beta$ trivially. Otherwise we have that $\varphi \models \alpha$ so $\alpha \wedge \varphi$ is equivalent to φ. But it is easy to see that φ does not have a minimal model, so again we have $\alpha \wedge \varphi \models^{\text{DC}} \beta$. However, φ is not downward persistent: it holds in the real numbers, but not in the substructure of the natural numbers. $\square$

Until now we have considered formulae that can be added to any premise, but we can also ask the question whether a formula respects monotonicity for a given, fixed premise.

Proposition 9.55 Given a preferential logic such that Mod is finite and for all $m \in \mathsf{Mod}$ there exists $\alpha^m \in \mathcal{L}$ such that $n \models \alpha^m$ if and only if $n = m$, let α be a fixed formula in $\mathcal{L}$. Then we have for all $\varphi \in \mathcal{L}$:

$$\forall\beta : (\alpha \models_{\preceq} \beta \Rightarrow \alpha \wedge \varphi \models_{\preceq} \beta) \Leftrightarrow \forall m \in \mathsf{Mod} : (m \models_{\preceq} \alpha \wedge \varphi \Rightarrow m \models_{\preceq} \alpha).$$

Proof: The right to left direction is trivial (and does not depend on the assumption). For the other direction, suppose that $\forall\beta : (\alpha \models_{\preceq} \beta \Rightarrow \alpha \wedge \varphi \models_{\preceq} \beta)$. Let $m \in \mathsf{Mod}$ be arbitrary and suppose $m \models_{\preceq} \alpha \wedge \varphi$. Now define $\beta = \bigvee\{\alpha^n \mid n \models_{\preceq} \alpha\}$; this is a well-defined formula since Mod was assumed finite. It is easy to see that $\alpha \models_{\preceq} \beta$: suppose $n \models_{\preceq} \alpha$, then α^n is one of the disjuncts of β, and by definition of α^n, we have $n \models \alpha^n$, so $n \models \beta$. But the assumption now gives that $\alpha \wedge \varphi \models_{\preceq} \beta$. As $m \models_{\preceq} \alpha \wedge \varphi$, we have $m \models \beta$, so there is an $n \in \mathsf{Mod}$ with $n \models_{\preceq} \alpha$ and $m \models \alpha^n$. But by definition of α^n this means that $m = n$ so $m \models_{\preceq} \alpha$. $\square$

Proposition 9.55 states that a formula φ respects monotonicity for a fixed premise α if and only if the minimal models of $\alpha \wedge \varphi$ are minimal models of α. Of course the criterion on the right-hand side is hard to check; we can give another criterion, but for that, we first need the following definition [KLM90]:

Definition 9.56 (Smoothness) A preferential logic is called *smooth*, if the following holds:

$$\forall\alpha \in \mathcal{L} : \forall m \in \mathsf{Mod} : (m \models \alpha \Rightarrow \exists n \in \mathsf{Mod} : n \preceq m \;\&\; n \models_{\preceq} \alpha).$$

This condition, which is also called *stopperedness* or *well-foundedness*, and is akin to the *limit assumption* of [Lew73], forbids chains of ever-decreasing models satisfying a formula. It is one of the basic properties in the framework of [KLM90].

Proposition 9.57 Given a smooth preferential logic, we have:

$$\forall m \in \mathsf{Mod} : (m \models_{\preceq} \alpha \wedge \varphi \Rightarrow m \models_{\preceq} \alpha)$$

if and only if

$$\forall m \in \mathsf{Mod} : (m \models \alpha \wedge \varphi \Rightarrow \exists n \in \mathsf{Mod} : (n \preceq m, n \models_{\preceq} \alpha \text{ and } n \models \varphi)).$$

The proof of this proposition is straightforward, and again it may not help much. As far as the examples are concerned, the conclusion of Proposition 9.55 holds for both Ground S5 and MTEL (the properties depend only on α and φ so we may restrict the signature and then use the proposition). Proposition 9.57 holds for Ground S5 (which is smooth). From these propositions we can find some sufficient conditions. If φ is downward persistent in the models of α, then Proposition 9.55 ensures that φ respects monotonicity with respect to α. If $\alpha \models_{\preceq} \varphi$ then Proposition 9.57 ensures that φ respects monotonicity with respect to α (but this also follows immediately with the rule of Cautious Monotonicity, which is satisfied in smooth preferential logics, [KLM90]). It seems hard to find a simple criterion necessary and sufficient for respecting monotonicity for a given premise. We leave this for further research.

9.2.4 Conservativity

In the previous subsection we have considered formulae that can always be added to a premise without invalidating any of the conclusions. In this subsection we will focus on the conclusions, and study formulae that, when they are concluded, can always be kept, no matter which new information is added to the premise. We will call these formulae conservative.

Definition 9.58 (Conservative) Given a preferential logic, we say a formula β is *conservative*, if

$$\forall \alpha, \varphi : \alpha \models_{\preceq} \beta \Rightarrow \alpha \wedge \varphi \models_{\preceq} \beta.$$

We have the following result connecting upward persistent and conservative formulae, in analogy with Proposition 9.44.

Proposition 9.59 Given a preferential logic that is smooth, if a formula is upward persistent, it is conservative.

Proof: Let β be upward persistent in a smooth preferential logic. Now suppose $\alpha \models_{\preceq} \beta$. Take any model m such that $m \models_{\preceq} \alpha \wedge \varphi$, then $m \models \alpha$ so by smoothness, there is a model n with $n \preceq m$ and $n \models_{\preceq} \alpha$. Then, as $\alpha \models_{\preceq} \beta$, we have $n \models \beta$. Since $n \preceq m$ and β is upward persistent, we have $m \models \beta$. This shows that $\alpha \wedge \varphi \models_{\preceq} \beta$, so β is conservative. $\square$

Again, we can ask if the upward persistent formulae are the only conservative formulae, and this is true under the same conditions as in the case of respecting monotonicity.

Proposition 9.60 (Only if ...) For a preferential logic that satisfies expressibility of preference we have: if a formula is conservative, then it is upward persistent.

Proof: Suppose β is not upward persistent, then there are $n, m \in \mathsf{Mod}$ such that $n \preceq m$, and $n \models \beta$ but $m \not\models \beta$. Now take $\alpha = \varphi^n$ and $\varphi = \varphi^m$. Then n is the only minimal model of α and $n \models \beta$ so $\alpha \models_{\preceq} \beta$, but m is a (actually, the only one) minimal model of $\alpha \wedge \varphi$, and $m \not\models \beta$, so $\alpha \wedge \varphi \not\models_{\preceq} \beta$. Thus, β is not conservative. □

Let us first identify the upward persistent formulae for our examples. This is relatively straightforward, since we have the following elementary result.

Proposition 9.61 For any preferential logic, φ is upward persistent if and only if $\neg\varphi$ is downward persistent.

This gives us the following.

Proposition 9.62

1. Define $BOX ::= K(\varphi) \mid BOX \wedge BOX \mid BOX \vee BOX \mid K(BOX)$ with φ propositional. Then a subjective S5-formula φ is upward persistent in Ground S5 if and only if it is S5-equivalent to a formula in BOX.
2. Define $TB ::= BOX \mid TB \wedge TB \mid TB \vee TB \mid F(TB) \mid G(TB) \mid P(TB) \mid H(TB)$. Then a subjective TEL-formula φ is upward persistent in MTEL if and only if it is equivalent (in the sense of $\sim_0$) to a formula in TB.
3. A first-order formula is upward persistent in predicate circumscription (of P) if and only if it is equivalent to a formula that is positive in P (meaning that all occurrences of the predicate P are in the scope of an even number of negations). A first-order formula is upward persistent in domain circumscription if and only if it is equivalent to an existential formula (a formula of the form $\exists x_1 \ldots x_n \psi$ where ψ is quantifier free).

Proof: Straightforward. □

In the above definition, formulae from BOX essentially only contain the K operator (the 'box' of S5); TB stands for 'temporal box' formulae. Now let us see what Propositions 9.59 and 9.60 say about the examples. Ground S5 satisfies expressibility of preference (for a finite language) and is smooth, so the conservative formulae are exactly the upward persistent formulae, which express only knowledge (and not ignorance). This can be lifted again to an infinite language. The fact that in Ground S5, formulae that express propositional knowledge, are conservative, was already noted in [DNR97]. MTEL also satisfies expressibility of preference (for a finite language), so any formula that is conservative, must be upward persistent, and must be equivalent to a formula in TB, expressing knowledge over time (not ignorance). This can be lifted to an infinite language. Unfortunately, MTEL is not smooth: the formula $F(Kp)$ is satisfiable, but has no minimal model. In MTEL, we have that $F(Kp) \models_{\preceq^g} F(Kq)$, but $F(Kp) \wedge Kp \not\models_{\preceq^g} F(Kq)$ ($F(Kp) \wedge Kp$ has a minimal model, in which only p is known, from the first point in time onwards). This means that the formula $F(Kq)$ is not conservative, although it is upward persistent. It is easy to see that in any preferential logic, valid formulae are always conservative, but in MTEL, these are (almost) the only ones, as will be shown below.

Definition 9.63 We call a TEL-model $\mathcal{M}$ *totally ignorant*, if for all propositional formulae φ we have: if $\mathcal{M} \models F(K\varphi)$ then φ is a propositional tautology. Define *the totally ignorant model* $\mathcal{M}^{ti}$ by $\mathcal{M}^{ti}(i) = \mathrm{Val}(P)$ for all i.

In a totally ignorant model, no knowledge is ever gained. *The* totally ignorant model is certainly *a* totally ignorant model, and if P is finite, it is the only one.

Proposition 9.64 For MTEL, in case P is infinite, we have that a formula is conservative if and only if it is true in all models. When P is finite, a formula is conservative if and only if it is true in all models except possibly the totally ignorant model.

Proof: We will prove that β is conservative if and only if it is true in all models that are not totally ignorant (both when P is finite and when it is infinite). First, suppose $\mathcal{M}$ is a model that is not totally ignorant, in which β is not true. So $\mathcal{M} \models F(K\gamma)$ for some propositional formula γ that is not a propositional tautology. If P is finite, we can consider the formula $\varphi^{\mathcal{M}}$ (see the proof of Proposition 9.53 for the definition of $\varphi^{\mathcal{M}}$). If P is infinite, one can show that $\mathcal{M}$ can be chosen in such a way that it is an 'inherently finite' model, allowing the construction of a formula $\varphi^{\mathcal{M}}$ with the same properties as for the case when P is finite (using essentially the same construction). The details of this argument are left to the reader. The formula $F(K\gamma)$ does not have a minimal model (γ must be known sometimes in the future, but this moment can always be postponed, yielding a smaller model), so $F(K\gamma) \models_{\preceq^g} \beta$. On the other hand, it can easily be shown that the only minimal model of $F(K\gamma) \wedge \varphi^{\mathcal{M}}$ is $\mathcal{M}$, which gives us $F(K\gamma) \wedge \varphi^{\mathcal{M}} \not\models_{\preceq^g} \beta$. This means that β is not conservative.

Now suppose that β is true in all models that are not totally ignorant, and suppose $\alpha \models_{\preceq^g} \beta$. Let $\mathcal{M}$ be a minimal model of $\alpha \wedge \varphi$. If $\mathcal{M}$ is not totally ignorant, then $\mathcal{M} \models \beta$. If it is totally ignorant, then also $\mathcal{M}^{ti} \models \alpha \wedge \varphi$ (it can be shown by induction that all totally ignorant models satisfy the same formulae). But then

$\mathcal{M}^{ti} \models \alpha$. Since no model is preferred over $\mathcal{M}^{ti}$, this means that $\mathcal{M}^{ti} \models_{\preceq} \alpha$ so $\mathcal{M}^{ti} \models \beta$, whence $\mathcal{M} \models \beta$. It follows that $\alpha \wedge \varphi \models_{\preceq^g} \beta$, so β is conservative.

Let us look at the case when P is infinite. Suppose $\mathcal{M} \not\models \beta$, with $\mathcal{M}$ totally ignorant. Now take a propositional atom p not occurring in β. It can easily be shown that we can find a model of Kp in which β is not satisfied. This model is of course not totally ignorant. This shows that if β is true in all models that are not totally ignorant, then it is true in all models. In case P is finite, $\mathcal{M}^{ti}$ is the only totally ignorant model. □

So in MTEL with an infinite P, valid formulae are the only conservative formulae. These formulae are of course upward persistent (in a trivial way), and they are equivalent to a formula in *TB*, for instance $K(\top)$. When the signature is finite, there are some extra formulae that are conservative, for example if $P = \{p, q\}$, then the formula $F(K(p \vee q) \vee K(\neg p \vee q) \vee K(p \vee \neg q) \vee K(\neg p \vee \neg q))$ is also conservative (it is true in all models except the totally ignorant one). Of course, this formula is upward persistent, and it is in *TB*.

Finite predicate and domain circumscription satisfy both expressibility of preference and smoothness, so the conservative formulae coincide with the upward persistent formulae which have the syntactic characterization of Proposition 9.62.

Full circumscription satisfies neither of the conditions.

Proposition 9.65 In predicate and domain circumscription there are upward persistent formulae that are not conservative.

Proof: First consider predicate circumscription. Let the language consist of three predicates besides equality, namely P, *Succ* (for 'successor') and $<$ (and P is circumscribed). Define the formulae α and φ as follows:

$$\begin{aligned}\alpha = \;& \forall x \exists ! y(Succ(x,y)) \wedge \\ & \forall x \exists ! y(Succ(y,x)) \wedge \\ & \forall xy(Succ(x,y) \to x < y) \wedge \\ & \forall xyz(x < y \wedge y < z \to x < z) \wedge \\ & \forall x(\neg(x < x)) \wedge \\ & \forall xy(x < y \vee y < x \vee x = y) \wedge \\ & \exists x Px \wedge \\ & \forall xy(Px \wedge Succ(x,y) \to Py)\end{aligned} \qquad \varphi = \forall xy(Px \wedge Succ(y,x) \to Py)$$

The intuitive meaning of α is that there are *Succ*-chains of elements, extending infinitely in both directions. If P occurs somewhere on such a chain, it must be true in all successors as well. A model of α can be made smaller (more preferred) by making P false in a point and all of its predecessors (leaving it true in all successors). We will now make this argument formal. The first claim is that α has no $\leq_P$-minimal models. Let M be a model of α. Then there must be an $x \in dom(M)$ with $x \in P^M$. Define $A = \{x\} \cup \{y \in P^M \mid (y, x) \in <^M\}$. Let N be the structure with the same domain as M, the same extension of $Succ$ and $<$, and $P^N = P^M \setminus A$. It is straightforward to verify that N is a model of α, and that $N \leq_P M$ and $N \neq M$.

On the other hand, $\alpha \wedge \varphi$ has minimal models. Let M be the structure with $dom(M) = \mathbb{Z}$ (the integers), $(a,b) \in Succ^M \Leftrightarrow b = a+1$, $(a,b) \in <^M \Leftrightarrow a < b$ in the natural ordering on the integers, and $P^M = \mathbb{Z}$. It can easily be checked that $M \models \alpha \wedge \varphi$. Now suppose $N \leq_P M$, $N \neq M$ and $N \models \alpha \wedge \varphi$. This means that $P^N \subset \mathbb{Z}$ (strict inclusion), and $P^N \neq \emptyset$ (as $N \models \exists x Px$). But then there must be $x, y \in \mathbb{Z}$ with $y = x + 1$, and either $x \in P^N$ and $y \notin P^N$, or $x \notin P^N$ and $y \in P^N$, contradicting either $N \models \forall xy(Px \wedge Succ(x,y) \to Py)$ ($N \models \alpha$) or $N \models \varphi$. Therefore M is a minimal model of $\alpha \wedge \varphi$.

Now define $\beta = \exists x(x \neq x)$, which is trivially upward persistent. Since α has no minimal models, we have $\alpha \models_P^{\mathrm{PC}} \beta$, but $M \not\models \beta$, so $\alpha \wedge \varphi \not\models_P^{\mathrm{PC}} \beta$. This shows that β is not conservative.

For domain circumscription, the example is quite similar. Again take $\beta = \exists x(x \neq x)$. Now define the formulae α and φ as follows:

$$\begin{aligned}\alpha = \quad & \forall x \exists! y(Succ(x,y)) \wedge \\ & \forall xy(Succ(x,y) \to x < y) \wedge \\ & \forall xyz(x < y \wedge y < z \to x < z) \wedge \\ & \forall x(\neg(x < x)) \wedge \\ & \forall xy(x < y \vee y < x \vee x = y) \wedge \\ & \forall xyz(Succ(x,z) \wedge Succ(y,z) \to x = y)\end{aligned} \qquad \varphi = \forall y \exists x(Succ(x,y))$$

One can now check that α has no $\leq_d$-minimal models, but $\alpha \wedge \varphi$ does, so the same β is upward persistent but not conservative in domain circumscription. The details are left to the reader. $\Box$

Until now, we have looked at formulae which, once concluded, are never lost, regardless of what new information comes in, but also regardless of what the initial premise was. However, we can also consider the situation with the premise fixed (analogously to the last part of the previous subsection): given a premise, which conclusions may be kept regardless of new information?

Proposition 9.66 For a preferential logic that satisfies expressibility of preference, if $\alpha \models_\preceq \beta$, then

$$(\forall \varphi : \alpha \wedge \varphi \models_\preceq \beta) \Leftrightarrow \alpha \models \beta.$$

Proof: Suppose $\alpha \models_\preceq \beta$.
"$\Leftarrow$" If $\alpha \models \beta$ then for any φ we have $\alpha \wedge \varphi \models \beta$ so $\alpha \wedge \varphi \models_\preceq \beta$.
"$\Rightarrow$" Suppose $\alpha \not\models \beta$, then there exists $m \in \mathsf{Mod}$ such that $m \models \alpha$ but $m \not\models \beta$. Then $m \models_\preceq \alpha \wedge \varphi^m$ (!), so $\alpha \wedge \varphi^m \not\models_\preceq \beta$. $\Box$

Note that the condition $\alpha \models_\preceq \beta$ was not used in the proof; if $\alpha \not\models_\preceq \beta$ then the equivalence is still true, as both sides are false. The proposition shows that the monotonic consequences of a premise are the only ones conservative with respect to this fixed premise.

Corollary 9.67 Let β be a conservative formula for a preferential logic that satisfies expressibility of preference, then $\alpha \models_{\preceq} \beta \Leftrightarrow \alpha \models \beta$.

Proof: If $\alpha \models \beta$ then in any preferential logic it follows that $\alpha \models_{\preceq} \beta$. On the other hand, if $\alpha \models_{\preceq} \beta$, then for any φ we have $\alpha \wedge \varphi \models_{\preceq} \beta$, since β is conservative. With Proposition 9.66 it follows that $\alpha \models \beta$. □

Given the usefulness of persistent formulae, we are interested in the complexity of determining persistence. We have some preliminary results.

Proposition 9.68 For a subjective TEL-formula φ it is decidable whether φ is downward persistent in MTEL. Similarly, it is decidable whether φ is upward persistent in MTEL.

Proof: First note that if downward persistence is decidable, then upward persistence is decidable as well, using Proposition 9.61. Furthermore, as persistence is independent of the propositional signature (which is straightforward to prove), we may assume the signature is finite. So suppose P contains n propositional atoms. We will prove that φ is downward persistent if and only if for all TELC-models $\mathcal{M}, \mathcal{N}$ with $\mathrm{size}(\mathcal{M}) \leq (2^n - 1) \cdot (2 \cdot \mathrm{depth}(\varphi) + 1)$, $\mathrm{size}(\mathcal{N}) \leq 2 \cdot (2^n - 1) \cdot (2 \cdot \mathrm{depth}(\varphi) + 1)$: if $\mathcal{N} \leq \mathcal{M}$ and $(\mathcal{M}, 0) \models \varphi$ then $(\mathcal{N}, 0) \models \varphi$. This implies the decidability of downward persistence.

Suppose φ is not downward persistent, then there exist TELC-models $\mathcal{M}, \mathcal{N}$ with $\mathcal{N} \prec^g \mathcal{M}$, $(\mathcal{M}, 0) \models \varphi$ and $(\mathcal{N}, 0) \not\models \varphi$. Now we construct a TELC-model $\mathcal{M}'$ by deleting points from sequences of more than $2 \cdot \mathrm{depth}(\varphi) + 1$ identical states before the stabilizing point from $\mathcal{M}$ until each such sequence is at exactly $2 \cdot \mathrm{depth}(\varphi) + 1$ states long. Then $\mathrm{size}(\mathcal{M}') \leq (2^n - 1) \cdot (2 \cdot \mathrm{depth}(\varphi) + 1)$, $\mathcal{N} \prec^g \mathcal{M}'$ and $(\mathcal{M}', 0) \models \varphi$ (by Lemma 9.13). Now we construct a model $\mathcal{N}'$ using the following procedure. First we identify all sequences of identical states in $\mathcal{N}$ after time point $(2^n - 1) \cdot (2 \cdot \mathrm{depth}(\varphi) + 1)$ but before the stabilizing point of $\mathcal{N}$ of length more than $(2 \cdot \mathrm{depth}(\varphi) + 1)$ points. From each such sequence we delete points until it has length $(2 \cdot \mathrm{depth}(\varphi) + 1)$. Then $\mathrm{size}(\mathcal{N}) \leq 2 \cdot (2^n - 1) \cdot (2 \cdot \mathrm{depth}(\varphi) + 1)$, $(\mathcal{N}, 0) \not\models \varphi$ (Lemma 9.13), and it is easily checked that $\mathcal{N}' \prec^g \mathcal{M}'$. □

This gives us another way of verifying TELC-theorems since $\vdash_{\mathbf{TELC}} \varphi \Leftrightarrow \mathcal{M}^{ti} \models \varphi$ and φ is upward persistent (note that for all TELC-models $\mathcal{N}$ we have $\mathcal{M}^{ti} \preceq^g \mathcal{N}$; then use soundness and completeness of **TELC**). Since TELC-theoremhood is co-NP-complete, we have as an immediate consequence:

Corollary 9.69 Upward persistence (and downward persistence) for subjective TEL-formulae is co-NP-hard.

For a valuation $m \in \mathrm{Val}(P)$ we can define the TELC-model $\mathcal{M}^m$ by $(\mathcal{M}^m)_t = \{m\}$ for all t. It is easy to see that such a model is maximal in the ordering $\preceq^g$, and this gives us another way of checking TELC-theorems since $\vdash_{\mathbf{TELC}} \varphi \Leftrightarrow \varphi$ is downward persistent and $(\mathcal{M}^m, 0) \models \varphi$ for all $m \in \mathrm{Val}(P)$. Furthermore we have: φ upward and downward persistent $\Leftrightarrow \vdash_{\mathbf{TELC}} at_0 \to \varphi$ or $\varphi \sim_0 \bot$, which gives us:

Corollary 9.70 Checking whether a subjective TEL-formula is both downward and upward persistent is co-NP-complete.

In the last two subsections, we have derived a number of results on formulae that respect monotonicity and conservative formulae and the links with persistent formulae. In the next subsection we will discuss the impact of these results in practice.

9.2.5 Practical implications

The results in this section may improve the efficiency of theorem provers for preferential logics, depending on a number of factors. In the first place, it is important how the theorem prover is used.

Consider the situation where we have a stand-alone theorem prover which gets different (unrelated) queries. Furthermore, suppose the theorem prover is asked to prove $\alpha_1 \wedge \ldots \wedge \alpha_n \models_\preceq \beta$. Then there are at least two possibilities for using the results in this section. First of all, suppose the preferential logic satisfies expressibility of preference. Then if β is upward persistent, we do not have to prove $\alpha_1 \wedge \ldots \wedge \alpha_n \models_\preceq \beta$, as it is equivalent to prove $\alpha_1 \wedge \ldots \wedge \alpha_n \models \beta$ (Corollary 9.67). In most preferential logics, preferential entailment is harder to compute than entailment in the underlying logic.

In the second place, sometimes local reasoning is possible (which is not possible in general for nonmonotonic logics): the theorem prover may derive the conclusion from part of the premise. So it may be the case that there is a $1 \leq k < n$ such that $\alpha_1 \wedge \ldots \wedge \alpha_k \models_\preceq \beta$ which is easier to verify than the original query. Then if $\alpha_{k+1}, \ldots, \alpha_n$ are downward persistent, Proposition 9.44 implies that $\alpha_1 \wedge \ldots \wedge \alpha_n \models_\preceq \beta$. If β is upward persistent (and the preferential logic is smooth), Proposition 9.59 sanctions $\alpha_1 \wedge \ldots \wedge \alpha_n \models_\preceq \beta$. For these results to be usable in a theorem prover, we need heuristic knowledge to decide if there is a promising split of the premise into two parts A and B. For such a split, we can then try to prove $A \models_\preceq \beta$ and downward persistence of B or upward persistence of β. In case $\alpha_1 \wedge \ldots \wedge \alpha_k \models_\preceq \beta$ can not be proved, we may have to directly prove $\alpha_1 \wedge \ldots \wedge \alpha_n \models_\preceq \beta$ after all.

These two methods will of course only improve efficiency if the determination of persistence is easier than the original query (we will treat this question below).

The second kind of situation is when we have a theorem prover which is used by an agent which has a lot of knowledge about the world, and from time to time performs observations to increase its knowledge. Then, although sometimes the agent will need to perform revisions, we are often in the situation that (many) conclusions from a premise (α) have been proved, and the premise is augmented by a new formula (φ). If this new formula is downward persistent, then the agent can retain all the old conclusions (and may only need to derive some new conclusions). If it is not, it can at least retain all the upward persistent conclusions (if the preferential logic is smooth). We may also try to determine if φ is downward persistent given α, or if β is conservative given α. Again, these methods only improve efficiency if it is easier to determine if φ respects monotonicity (possibly given α) than recomputing all old conclusions, or if it is easier to determine that β is conservative (possibly given α) than checking $\alpha \wedge \varphi \models_{\preceq} \beta$.

The possible efficiency improvement in both cases heavily depends on the cost of determining persistence relative to the cost of determining preferential consequence. Unfortunately, it is very hard to say anything about this issue in general. It depends on the preferential logic at hand, on the representation of the logic (syntactically, as a proof calculus, or semantically, as models with a preference relation), and on other implementation issues. For instance, it can be important how much information is retained from previous queries: whether proofs or minimal models are stored. Let us consider the examples again.

Preferential entailment in both Ground S5 and MTEL is Π_3^{P}-complete, whereas full circumscription is undecidable (restricted versions of circumscription exist which are decidable, but still highly complex). Unfortunately, determining downward or upward persistence is not easier for these logics. We have seen, however, that the classes of persistent formulae have syntactic representations of the form: φ is upward/downward persistent if and only if it is equivalent to a formula in C, where C is a (syntactic) class of formulae. Now, of course, determining *equivalence to* a formula in C is as complex as determining persistence, but there may be subclasses of a class of persistent formulae, with a lower complexity. For instance, determining *membership* of C is much easier, namely polynomial. The members of C are persistent. So what we propose is to check membership of C, instead of equivalence to a member of C. In that case, we will miss some persistent formulae (and have to prove the original query), but this disadvantage is outweighed by the complexity advantage of checking membership. The checking of membership can be improved upon by adding some (easy) checks for equivalence to a formula in C. For instance, in Ground S5, if we consider, for a formula φ, for each propositional subformula, the nearest K operator in which scope it lies, then if all of these K operators are in the scope of an odd number of negations, we can conclude that φ is downward persistent. The formula $\neg K(q \vee Kp)$, for example, satisfies this condition, and although it is not a member of $DIAM$, it is equivalent to $\neg Kq \wedge \neg Kp \in DIAM$. This check is obviously polynomial.

Given a preferential logic, the designer of a theorem prover could proceed as follows. First, syntactic classes of formulae that are downward and upward persistent have to be identified. For Ground S5, MTEL and predicate and domain circumscription, these can be found in Definitions 9.45, 9.47, 9.49 and Proposition 9.62.

For other preferential logics, if such classes are trivial (they may, for instance, only include tautologies and contradictions), then the usefulness of the results is limited. Otherwise, the theorem prover could work as follows. Given a query of the form $\alpha \wedge \varphi \models_{\preceq} \beta$, first it is checked if φ belongs to the syntactic class of downward persistent formulae or if β belongs to the syntactic class of upward persistent formulae (this latter test should only be performed if the preferential logic is smooth). If β is conservative and the logic satisfies expressibility of preference, it tries to prove $\alpha \wedge \varphi \models \beta$ (this usually has a lower complexity than the original query; for Ground S5 and MTEL, we saw in the previous section that monotonic consequence is NP-complete). The answer of this query is the answer to the original query (see Corollary 9.67). Otherwise, if φ belongs to the syntactic class of downward persistent formulae or if β belongs to the syntactic class of upward persistent formulae (but the logic does not satisfy expressibility of preference), then the theorem prover tries to prove $\alpha \models_{\preceq} \beta$. If this succeeds, it outputs yes. Otherwise, it will try to answer the original query directly.

As stated before, the practical savings partly depend on representation and implementation aspects. It also depends on the application domain and use of the theorem prover: if formulae in these syntactic classes occur often, the efficiency improvement is higher than if they are infrequent.

9.2.6 Conclusions and related work

The results in this section may lead to more efficient implementations of preferential logics. Experimenting with theorem provers which use these results is necessary in order to determine the efficiency improvement in practice.

It would be nice to find a better characterization of formulae that can be added to a given, fixed premise without destroying conclusions.

Syntactic characterizations of persistent formulae were given for a number of example preferential logics, but we would like to have a result for broader classes of preferential logics, such as the class of ground nonmonotonic modal logics (see [DNR97]). In that publication it is already stated that in Ground S5, formulae that express propositional knowledge are conservative.

9.3 Non-cumulative reasoning: rules and models

In the previous section, we looked at monotonicity for subclasses of formulae in preferential logics. This was inspired by the fact that our logic MTEL, a preferential logic, is nonmonotonic, and we indeed found classes of formulae for which monotonicity holds. Another variant of monotonicity we already mentioned, is Cautious Monotonicity (CM, see [KLM90]). This rule states that any formula can be added to the premises without invalidating earlier conclusions, if it follows from these premises:

If $\alpha \mid\!\sim \gamma$ and $\alpha \mid\!\sim \beta$ then $\alpha \wedge \beta \mid\!\sim \gamma$ (*Cautious Monotonicity*)

This rule is not satisfied by MTEL: if we take $\alpha = F(Kp)$, $\beta = Kp$ and $\gamma = \bot$, then α has no $\preceq^g$-minimal model, so that by definition we get $\alpha \models_{\preceq^g} \gamma$ and $\alpha \models_{\preceq^g} \beta$. However, $\alpha \wedge \beta$ has a minimal model (in which p and its consequences are known from time point zero onwards, and nothing else), in which γ is obviously not true, so $\alpha \wedge \beta \not\models_{\preceq^g} \gamma$.

The idea of preferential logics, with standard models augmented with a preference relation, has been quite influential, and many variations on Shoham's semantics [Sho87, Sho88] have been proposed and studied. Probably one of the most well-known studies dealing with this subject is [KLM90]. In that paper it is shown that for a number of classes of nonmonotonic logics, semantics can be defined using models with states and a preference ordering. For different classes, various restrictions can be placed on such models (see also [GM94a], [Mak89], [Mak94], and e.g. [Sch92b]). The logics considered in [KLM90], however, all satisfy the rule of Cautious Monotonicity. In this section, we will try to complement the work of [KLM90] by performing a similar analysis, but for logics that need not satisfy CM.

9.3.1 Cautious Monotonicity and smoothness

The condition on the preferential semantics that ensures that the logic satisfies Cautious Monotonicity, is the following:

> for every formula α and for every state s satisfying α, either s is minimal among all states satisfying α, or there exists a more preferred state t satisfying α, which is minimal among all states satisfying α.
> (Smoothness)

Notice that MTEL does not satisfy this condition: if $\alpha = FKp$, and we have any model for it, then there is always a smaller model of α, so that it does not have any minimal models. Smoothness, also called *well-foundedness* or *stopperedness*, is similar to the *Limit Assumption* ([Lew73]), and the *Uniqueness Assumption* ([Sta68]) used in semantics for conditional logic (see e.g., [Nut84]), which state that for any formula α there should be a (respectively a unique) state which is minimal among all states satisfying α. In conditional logic, it is widely acknowledged that both of these assumptions are highly suspect from an ontological point of view ([Nut84, Bou94a]). Consider the example (similar to the one in [Lew73]) of a premise α which says: "Pete is over eight feet tall". Since we know that people are rarely over eight feet tall, among states in which Pete is over eight feet tall, we prefer a state s to a state t whenever Pete's height in s is less than his height in t. But then there can surely be no most preferred state where α holds: for each state where α holds, there is always another state in which Pete's height is less, but still over eight feet. So there is a conflict between wanting to retain Cautious Monotonicity on the one hand, and the undesirability of conditions like smoothness on the other hand.

9.3.2 Resolving the conflict

If one is committed to using some sort of preferential semantics for nonmonotonic logics, there are a number of ways to deal with the aforementioned conflict. One solution is to impose smoothness anyway: by not mentioning any problems (as is done in [KLM90]), by stating that it is ugly but necessary, or even by denying the possibility of infinite sequences of ever more preferred states.

Recently a number of researchers have proposed a different definition of preferential entailment (e.g. [Bou94a, Bou94b]), which validates CM without the need for smoothness. When minimal α-states exist, it is equivalent to the normal definition, but infinite sequences of more and more preferred states are handled differently (see [Bou94a] for details). Although this approach is certainly viable, it has the disadvantage that the definition of entailment becomes more difficult, and has less intuitive clarity.

An obvious third candidate, which has nevertheless not been investigated thoroughly, is to reject Cautious Monotonicity. In [KLM90] it is argued (but also by others, e.g. Gabbay), that a system which does not satisfy CM (and some other basic properties we will mention later), should not be considered a logical system. But even there it is said: "This appreciation probably only reflects the fact that, so far, we do not know anything interesting about weaker systems" ([KLM90], p. 176). However, many interesting weaker logics exist: default logic [Rei80b], circumscription [McC80] (when infinite models are also considered), and of course our logic MTEL. There are also some intuitive practical examples in which CM should not hold: see for instance [ZR97], or [Vre92] which argues against CM. In [Voo93] it is argued that the fact that default logic does not satisfy CM is not a disadvantage of default logic, but that rational agents (and [Voo93] also states that not all agents need to be perfectly rational) should not adopt a set of defaults giving rise to a non-cumulative consequence relation (just as rational agents should not adopt an inconsistent set of beliefs).

One situation in which CM may be violated is with 'nonmonotonically inconsistent' premises. In [KLM90], when we have $\alpha \mid\!\sim \bot$, this is because α is inconceivable (a statement like "I am the queen of England" uttered by myself), and there should be no state in the preferential model where α holds. In MTEL, however, a formula like FKp does not denote an inconceivable state of affairs, but it is nonmonotonically inconsistent in that it is not preferentially satisfiable.

One explanation of the fact that CM intuitively sounds good, whereas there are systems which do not satisfy it, lies in the reading of the symbol $\mid\!\sim$: often, $\alpha \mid\!\sim \beta$ is read as "if α then normally β". But there are many other possible readings: "if α then β is a reasonable assumption", "If α is all I know, then β" or "if α holds then β holds in most cases". In all of these readings, examples can be constructed where CM should not intuitively hold. More in general, intuition does not provide a very firm basis for judging logics (see the first chapter of [Vel85] for an extensive discussion of this point). When accepting rules on the basis of intuitive examples, we quickly end up with systems satisfying monotonicity after all.

As mentioned earlier, in this section we shall attempt to perform an analysis similar in spirit to the one of [KLM90], but for logics that do not necessarily satisfy CM. The selection of rules we shall consider, and the order in which they are treated, are partly inspired by [KLM90], partly by the rules satisfied by MTEL. We shall start with four basic rules in Subsection 9.3.3, for which we will give a representation result in Subsection 9.3.4. In Subsection 9.3.5 we will add the rule “Cut”, in Subsection 9.3.6 the rule “Or”, and in Subsection 9.3.7 the rule “Weak Rational Monotonicity”. Subsection 9.3.8 discusses other rules and Subsection 9.3.9 gives some final remarks.

9.3.3 The four basic rules

The most basic rule which should hold for an inference relation $\mid\!\sim$, as opposed to relations that describe some sort of revision, is reflexivity. (When stating a rule, we will leave out quantification; all formulae appearing in the rules are universally quantified. The name of the rule appears between brackets; for ease of reference we will also number them)

$$\alpha \mid\!\sim \alpha \qquad \text{(Rule 1: Reflexivity)}$$

From this point on, we shall make the restriction that all inference relations are built ‘on top of’ a classical logic (formally, this will mean that they satisfy the rules 1-4). Concerning this classical, underlying logic, we shall make the following assumptions, corresponding to those in [KLM90]:

Assumption 9.71

- we have a language, $\mathcal{L}$, of well-formed formulae, closed under the classical propositional connectives.
- the semantics of this language is given by a set $\mathcal{U}$, the elements of which will be called *worlds*, and a binary relation $\models$ of satisfaction between worlds and formulae, which satisfies, for $\alpha, \beta \in \mathcal{L}, u \in \mathcal{U}$:
  - $u \models \neg\alpha$ iff $u \not\models \alpha$, and
  - $u \models \alpha \vee \beta$ iff $u \models \alpha$ or $u \models \beta$.

  By $\models \alpha$ we denote that $u \models \alpha$ holds for all $u \in \mathcal{U}$.
- we have compactness: a set of formulae is satisfiable if all of its finite subsets are.

Except for compactness, these assumptions are the same as Assumption 9.42. Remember that MTEL (and also the altered version of Section 9.2) are *not* compact. Therefore, the results of this section do not apply to MTEL, although this section and our interest in logics that do not satisfy CM was originally inspired by MTEL. The rules we shall treat in the sequel (Rules 1 through 7) are all satisfied by the altered version of MTEL.

The next two rules concern the interaction with the underlying logic:

If $\alpha \mathrel{|\!\sim} \beta$ and $\models \beta \rightarrow \gamma$ then $\alpha \mathrel{|\!\sim} \gamma$ (*Rule 2: Right Weakening*)

If $\models \alpha \leftrightarrow \beta$ then ($\alpha \mathrel{|\!\sim} \gamma$ iff $\beta \mathrel{|\!\sim} \gamma$) (*Rule 3: Left Logical Equivalence*)

The first rule states that if β is a consequence of α and γ holds whenever β holds, then γ should be a consequence of α too. The second rule states that only the meaning of the premise should matter, not its syntactical form. These three rules are also in the basic five rules of [KLM90], along with Cut and Cautious Monotonicity. We will treat Cut in the next subsection, and, as said before, we explicitly do not want to consider CM. Instead, we will take the following rule, which follows from the basic five rules of [KLM90], as basic:

If $\alpha \mathrel{|\!\sim} \beta$ and $\alpha \mathrel{|\!\sim} \gamma$ then $\alpha \mathrel{|\!\sim} \beta \wedge \gamma$ (*Rule 4: And*)

This rule states that the conjunction of two consequences of a formula is also a consequence. There exist systems in which this rule is not satisfied: credulous entailment in default logic (see [Poo89]; sceptical entailment does satisfy And). This rule causes the Lottery Paradox: in a lottery, since each ticket may have only a small chance of winning the prize, we could infer for every ticket that it will (normally) not win the prize. But if we take the conjunction of all these statements, we end up with the (obviously undesirable) conclusion that normally no ticket will win the prize. Still many interesting systems satisfy it, and we will adopt it here.

For ease of reference, we shall now list and explain some of the rules that will be needed and discussed later. For a more thorough explanation and for intuitive examples of why these rules should hold, we refer the reader to [KLM90]. In these rules, $\mathrel{|\!\sim}\alpha$ is an abbreviation for $\top \mathrel{|\!\sim} \alpha$, where $\top$ is an abbreviation for any tautology.

If $\alpha \mathrel{|\!\sim} \beta$ and $\alpha \wedge \beta \mathrel{|\!\sim} \gamma$ then $\alpha \mathrel{|\!\sim} \gamma$ (*Rule 5: Cut*)

If $\alpha \mathrel{|\!\sim} \gamma$ and $\beta \mathrel{|\!\sim} \gamma$ then $\alpha \vee \beta \mathrel{|\!\sim} \gamma$ (*Rule 6: Or*)

If $\alpha \wedge \beta \mathrel{|\!\sim} \gamma$ then $\alpha \mathrel{|\!\sim} \beta \rightarrow \gamma$ (*Conditionalization*)

If $\alpha \mathrel{|\!\sim} \beta$ then $\mathrel{|\!\sim}\alpha \rightarrow \beta$ (*Weak Conditionalization*)

If $\not\mathrel{|\!\sim}\neg\alpha$ and $\mathrel{|\!\sim}\alpha \rightarrow \beta$ then $\alpha \mathrel{|\!\sim} \beta$ (*Rule 7: Weak Rational Monotonicity*)

The first rule, Cut, expresses that in order to prove a conclusion from a premise, one may add a hypothesis to the premise and derive the conclusion from this enlarged premise, after which one has to prove that this hypothesis in fact follows from the original premise. This rule is one of the basic rules of [KLM90], and we will consider it in Section 9.3.5.

The rule Or states that if a conclusion follows independently from two premises, it should also follow from their disjunction. This rule is also covered in [KLM90], and we shall treat this rule in Section 9.3.6.

The rule Conditionalization (called S in [KLM90]) corresponds to the deduction theorem (or one half of the deduction theorem according to some books) in classical logics. It expresses the possibility of proving an implication from a premise by proving the consequent from the premise augmented with the antecedent. Weak Conditionalization is (obviously) a weaker variant of Conditionalization, in which the premise must be a tautology.

The last rule we shall consider is Weak Rational Monotonicity, which gives a partial converse of Weak Conditionalization. It states that in order to prove a formula from a premise, one can prove the corresponding implication from the empty premise, provided that the negation of the original premise can not be proved from the empty premise. The reason we shall consider this rule (in Section 9.3.7), is that entailment in MTEL satisfies this rule and all rules considered up till now (Rules 1-7, and (Weak) Conditionalization), and none of the other rules of [KLM90] which can not be proved from these rules (this is left as an exercise for the reader).

Not all of the above rules are independent. We shall give a few dependencies. Conditionalization (and thus also Weak Conditionalization) is implied by the combination of Reflexivity, Right Weakening , Left Logical Equivalence, and Or ([KLM90], p. 191). Furthermore, And, Right Weakening, and Conditionalization imply Cut ([KLM90], p. 191). We shall not treat Conditionalization and Weak Conditionalization separately, but focus on Rules 1-7.

We shall begin with a representation result for the four basic rules.

9.3.4 Representation for the four basic rules

The rules of Reflexivity, And, Left Logical Equivalence, and Right Weakening imply that the set of consequences of a formula α is a deductively closed set of formulae containing α, and that formulae equivalent to α have the same set of consequences as α.

We can give semantics to such consequence relations, borrowing from the theory of nonmonotonic model operators (see e.g. [DH94]). In a way analogous to the techniques in [KLM90] we define inference models to base consequence relations on.

Definition 9.72 (Inference models) An *inference model* W is a triple $\langle S, \ell, P\rangle$ where S is a set (the set of *states*), ℓ is a function $\ell : S \to \mathcal{U}$ which assigns a world to every state and P is a function $P : \mathcal{L} \to \mathcal{P}(S)$ which assigns to every formula a set of states (the *intended* states) such that:

1. $P(\alpha) \subseteq \{s \in S \mid \ell(s) \models \alpha\}$, and

2. $\models \alpha \leftrightarrow \beta$ implies $P(\alpha) = P(\beta)$.

Note that there may be two different states which are labeled with the same world. Although this is not necessary (we could have defined an inference model as a tuple $\langle S, P\rangle$ where S is a subset of $\mathcal{U}$) for the representation result (Proposition 9.76), we need labeled states for later representation results. For uniformity, we have used separate states and a labeling function already here. When using preference relations later on, we will need states that are labeled with *sets* of worlds instead of single worlds. It is possible to define the labeling function in such a way for inference models, but that would make proofs needlessly complicated.

Definition 9.73 (Consequence relation of an inference model) Given an inference model $W = \langle S, \ell, P\rangle$ we define the associated consequence relation $\mid\!\sim_W$ by: $\alpha \mid\!\sim_W \beta$ iff $P(\alpha) \models \beta$, where $P(\alpha) \models \beta$ iff for all $s \in P(\alpha)$ we have $\ell(s) \models \beta$.

Before we prove a representation theorem, we need a definition from [KLM90]:

Definition 9.74 (Normal world) A world $u \in \mathcal{U}$ is called a *normal world for* α with respect to a consequence relation $\mid\!\sim$ if for all $\beta \in \mathcal{L}$ for which $\alpha \mid\!\sim \beta$ we have $u \models \beta$.

So a normal world for α satisfies all consequences of α. A crucial lemma for many of the representation theorems is the following (Lemma 3.18 from [KLM90]):

Lemma 9.75 If $\mid\!\sim$ satisfies And, Reflexivity and Right Weakening then all normal worlds for α satisfy β if and only if $\alpha \mid\!\sim \beta$.

Now we will show the representation result for consequence relations satisfying the above mentioned four basic rules (Rules 1-4):

Proposition 9.76 (Representation for inference models) A consequence relation $\mid\!\sim$ satisfies Reflexivity, And, Left Logical Equivalence, and Right Weakening if and only if $\mid\!\sim \; = \; \mid\!\sim_W$ for some inference model W.

Proof: The proof from right to left is easy and left to the reader. For the other direction define $W = \langle S, \ell, P\rangle$ with $S = \mathcal{U}$, the labeling function ℓ is the identity function and for all $\alpha \in \mathcal{L}$: $P(\alpha) = \{m \in \mathcal{U} \mid m$ is a normal world for α with respect to $\mid\!\sim\}$. By Reflexivity we have that $\alpha \mid\!\sim \alpha$ so $P(\alpha) \subseteq \{s \in S \mid \ell(s) \models \alpha\}$. Furthermore, by Left Logical Equivalence we have that if $\models \alpha \leftrightarrow \beta$ then ($\alpha \mid\!\sim \gamma$ iff $\beta \mid\!\sim \gamma$) so a world m is a normal world for α iff it is a normal world for β, so $P(\alpha) = P(\beta)$. Thus, W is an inference model. It follows immediately from the definitions and Lemma 9.75 that for all $\alpha, \beta \in \mathcal{L}$: $\alpha \mid\!\sim \beta$ if and only if $\alpha \mid\!\sim_W \beta$. $\square$

Now we want to look at consequence relations which satisfy in addition to the above four rules also the rule Cut, and we will see that we can get a similar representation result, but also one which uses a preference relation.

9.3.5 Adding the rule "Cut"

A representation result for Rules 1-5 using model operators (similar to Proposition 9.76) is easily obtained:

Proposition 9.77 A consequence relation $\mid\sim$ satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Cut if and only if $\mid\sim = \mid\sim_W$ for some inference model W which satisfies: if $P(\alpha) \models \beta$ then $P(\alpha) \subseteq P(\alpha \wedge \beta)$.

Proof: Straightforward (use the same definition as in the proof of Proposition 9.76). $\square$

What we really want is not a model operator, but a preference relation which singles out the most preferred worlds. We will show that this is possible, using the most liberal notion of model, which is the same as a cumulative model in [KLM90], but without the smoothness condition defined in the introduction. We will first give a definition of such a model and its associated model operator and consequence relation.

Definition 9.78 (Ordered model) An *ordered model* W is a triple $\langle S, \ell, \prec \rangle$ where S is a set (the set of *states*), ℓ is a function $\ell : S \rightarrow \mathcal{P}(\mathcal{U})$ which assigns a set of worlds to every state and $\prec$ is a binary relation on S.

Definition 9.79 (Consequence relation of an ordered model) Given an ordered model $W = \langle S, \ell, \prec \rangle$ we define:

- the model operator P_W: $P_W(\alpha) = \min_{\prec}\{s \in S \mid \ell(s) \models \alpha\}$, where $\min_R A = \{a \in A \mid \text{there exists no } b \in A \text{ such that } bRa\}$. The states in $P_W(\alpha)$ are called *α-minimal* states (with respect to W). If $\alpha = \top$, these states are just called *minimal*.

- the associated consequence relation $\mid\sim_W$: $\alpha \mid\sim_W \beta$ iff $P_W(\alpha) \models \beta$. Here $P_W(\alpha) \models \beta$ means that $\ell(s) \models \beta$ for all $s \in P_W(\alpha)$, and $\ell(s) \models \beta$ means that $m \models \beta$ for all $m \in \ell(s)$.

We want to prove that the class of consequence relations of ordered models is exactly the class of consequence relations which satisfy the basic four rules (Rules 1-4) and the Cut rule. First we show that the consequence relation of an ordered model satisfies the five rules:

Lemma 9.80 (Soundness) For any ordered model W, its associated consequence relation satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Cut.

Proof: These properties are all straightforward; as an example we will check Cut. So suppose that $\alpha \mathrel{|\!\sim}_W \beta$ and $\alpha \wedge \beta \mathrel{|\!\sim}_W \gamma$. Take any α-minimal state s. Then $\ell(s) \models \beta$ and obviously $\ell(s) \models \alpha$, so $\ell(s) \models \alpha \wedge \beta$, but it is also $\alpha \wedge \beta$-minimal: if $t \prec s$ and $\ell(t) \models \alpha \wedge \beta$ then in particular $\ell(t) \models \alpha$. This is impossible, since s was an α-minimal state. Therefore, we have that $\alpha \mathrel{|\!\sim}_W \gamma$ which proves Cut. □

We now intend to show that for any consequence relation satisfying the five rules (Rules 1-5) we can define an ordered model with a consequence relation identical to the one we started with:

Definition 9.81 Given a consequence relation $\mathrel{|\!\sim}$ which satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Cut we define its *associated ordered model* $W = \langle S, \ell, \prec\rangle$ by:

- $S = \{(X, \alpha) \mid X \subseteq \mathcal{U}, \alpha \in \mathcal{L}\}$,
- $\ell((X, \alpha)) = X$, and
- $(X,\alpha) \prec (Y,\beta)$ iff $\begin{cases} Y \models \alpha, \\ Y \text{ contains a world not normal for } \alpha \text{ with respect to } \mathrel{|\!\sim}, \\ X \models \alpha, \text{ and} \\ \text{if } Y = \{m \in \mathcal{U} \mid m \text{ is normal for } \beta \text{ with respect to } \mathrel{|\!\sim}\} \\ \quad \text{then } X \not\models \beta. \end{cases}$

The intuition behind this definition is as follows. We want the preference relation to select only sets of worlds which are all normal for α (we want to use Lemma 9.75). So whenever a state (Y, β) satisfying α contains a world which is not normal for α, there must be a state satisfying α below it (in the ordering). Such a state (X, α) can be considered as a 'witness' for the non-normality with respect to α of (Y, β) (therefore X is 'labeled' with α). But at the same time, if Y is the set of all normal β worlds, we want it to be selected by the preference relation, so any state below it should not satisfy β. So here the 'labeling' of Y with β serves another purpose: this is the state (for β) which should be selected.

We want to show that the consequence relation based on this model is identical to the one we started with:

Lemma 9.82 (Completeness) For a consequence relation $\mathrel{|\!\sim}$ which satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Cut, its associated ordered model W induces a consequence relation $\mathrel{|\!\sim}_W$ with $\mathrel{|\!\sim} = \mathrel{|\!\sim}_W$.

Proof: We will first show that

$$\bigcup \ell[P_W(\alpha)] = \{m \in \mathcal{U} \mid m \text{ is normal for } \alpha \text{ with respect to } \mid\!\sim\}.$$

$\subseteq$: Suppose m is a world which is not normal for α. Furthermore, suppose that $m \in Y$ for some state (Y, β) with $Y \models \alpha$. We consider two cases:

- $Y \neq \{m \in \mathcal{U} \mid m$ is normal for β with respect to $\mid\!\sim\}$. Then we have $(Y, \alpha) \prec (Y, \beta)$ so $(Y, \beta) \notin P_W(\alpha)$.
- $Y = \{m \in \mathcal{U} \mid m$ is normal for β with respect to $\mid\!\sim\}$. Remember that with Lemma 9.75 we have that for all γ, $Y \models \gamma \Leftrightarrow \beta \mid\!\sim \gamma$. As m is not normal for α, there must be a formula γ such that $\alpha \mid\!\sim \gamma$ but $m \not\models \gamma$. We would like to show that a world $n \in \mathcal{U}$ exists such that $n \models \alpha$ and $n \not\models \beta$. To this end we will show that $\not\models \alpha \to \beta$; So suppose by contradiction that $\models \alpha \to \beta$. Then we have that $\models \alpha \leftrightarrow (\beta \wedge \alpha)$, and with Left Logical Equivalence this gives $\beta \wedge \alpha \mid\!\sim \gamma$. As $Y \models \alpha$ also $\beta \mid\!\sim \alpha$ and then Cut yields $\beta \mid\!\sim \gamma$. Because $m \in Y$, it is normal for β so $m \models \gamma$ contradicting the assumption. So we do not have that $\models \alpha \to \beta$ and therefore there exists $n \in \mathcal{U}$ with $n \models \alpha$ and $n \not\models \beta$. It can be easily checked that $(\{n\}, \alpha) \prec (Y, \beta)$ so $(Y, \beta) \notin P_W(\alpha)$.

We have proved that $m \notin \bigcup \ell[P_W(\alpha)]$.

$\supseteq$: Define $Y = \{m \in \mathcal{U} \mid m$ is normal for α with respect to $\mid\!\sim\}$ and consider the state (Y, α). By Reflexivity, $\alpha \mid\!\sim \alpha$ so $Y \models \alpha$. Suppose $(X, \beta) \prec (Y, \alpha)$, then by definition $X \not\models \alpha$. We have proved that $(Y, \alpha) \in P_W(\alpha)$.

To conclude our argument, note that $\alpha \mid\!\sim_W \beta$ iff $\bigcup \ell[P_W(\alpha)] \models \beta$ iff $\{m \in \mathcal{U} \mid m$ is normal for α with respect to $\mid\!\sim\} \models \beta$ iff $\alpha \mid\!\sim \beta$. □

Theorem 9.83 (Representation for ordered models) A consequence relation $\mid\!\sim$ satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Cut if and only if $\mid\!\sim \,=\, \mid\!\sim_W$ for some ordered model $W = \langle S, \ell, \prec \rangle$.

Proof: Follows immediately from Lemma 9.80 and Lemma 9.82. □

We can restrict the relation $\prec$ to an irreflexive one. Each reflexive point s in an ordered model $W = \langle S, \ell, \prec \rangle$ can be replaced by an infinite sequence (s_i) of states labeled with $\ell(s)$ and $s_i \prec s_{i+1}$. Each element $s \prec t$ has to be replaced by $s_i \prec t$ for all i, and the same for elements $t \prec s$. It is easy to see that $\mid\!\sim_W$ is not affected by these changes.

We would like to find a similar representation result when the rule Or is included, which will be done in the next subsection.

9.3.6 Adding the rule "Or"

We would like to have a representation result when we add the rule Or to our set of five rules (Rules 1-5). As the basic four rules (Rules 1-4) together with Or imply Cut, we will no longer mention that rule. For the representation, we will first need a small lemma:

Lemma 9.84 If a consequence relation $\mid\!\sim$ satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Or, then for any α, β and γ: if $\models \alpha \to \beta$ and $\alpha \mid\!\sim \gamma$ then $\beta \mid\!\sim \alpha \to \gamma$.

Proof: From $\models \alpha \to \beta$ we have $\models \alpha \leftrightarrow (\beta \wedge \alpha)$ and with Left Logical Equivalence we have $\beta \wedge \alpha \mid\!\sim \gamma$. Using Conditionalization (which follows from the other rules), we get $\beta \mid\!\sim \alpha \to \gamma$. □

Proposition 9.85 A consequence relation $\mid\!\sim$ satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Or if and only if $\mid\!\sim = \mid\!\sim_W$ for some inference model $W = \langle S, \ell, P \rangle$ with P such that for all α, β: if $\models \alpha \to \beta$ then $P(\beta) \cap \{s \in S \mid \ell(s) \models \alpha\} \subseteq P(\alpha)$.

Proof:

- Going from right to left we have to prove that $\mid\!\sim_W$ satisfies Or: suppose $\alpha \mid\!\sim_W \gamma$ and $\beta \mid\!\sim_W \gamma$. Choose $s \in P(\alpha \vee \beta)$. Then $\ell(s) \models \alpha \vee \beta$, so $\ell(s) \models \alpha$ or $\ell(s) \models \beta$. Suppose $\ell(s) \models \alpha$. As $\models \alpha \to (\alpha \vee \beta)$ we have $s \in P(\alpha)$ by the right-hand condition and as $\alpha \mid\!\sim_W \gamma$ we have $\ell(s) \models \gamma$. So $P(\alpha \vee \beta) \models \gamma$, and by definition $\alpha \vee \beta \mid\!\sim_W \gamma$ as required.
- For the other direction we take the same definition for W as in Propositions 9.76 and 9.77, and we only have to prove the extra condition. So suppose $\models \alpha \to \beta$ and take $m \in P(\beta) \cap \{s \in S \mid \ell(s) \models \alpha\}$. Suppose that $\alpha \mid\!\sim \gamma$. Then Lemma 9.84 gives us that $\beta \mid\!\sim \alpha \to \gamma$ and as m is a normal world for β by the definition of P, we have $m \models \alpha \to \gamma$, but as $m \in \{s \in S \mid \ell(s) \models \alpha\}$ we have $m \models \alpha$ and so $m \models \gamma$. We have proved that m is a normal α world, so $m \in P(\alpha)$.

□

Note that the condition in Proposition 9.85 indeed implies the condition in Proposition 9.77. As before, we would also like a representation result using preference relations.

Definition 9.86 (Sv-model) A *single-valued ordered model* (or *sv-model*) W is a triple $\langle S, \ell, \prec \rangle$ where S is a set (the set of *states*), ℓ is a function $\ell : S \to \mathcal{U}$ which assigns a world to every state and $\prec$ is a binary relation on S. The model operator P_W is defined as in Definition 9.79, and the associated consequence relation $\mathrel{|\!\sim}_W$ is also defined as in Definition 9.79, viewing $\ell(s)$ as a singleton set.

We want to prove that the class of consequence relations of sv-models is exactly the class of consequence relations which satisfy the basic four rules (Rules 1-4) and the Or rule. First we show that the class of consequence relations of an sv-model satisfies the five rules:

Lemma 9.87 (Soundness) For any sv-model W, its associated consequence relation satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Or.

Proof: This is again straightforward. $\square$

We now intend to show that for any consequence relation satisfying the five rules (Rules 1-4 and 6) we can define an sv-model with a consequence relation identical to the one we started with:

Definition 9.88 Given a consequence relation $\mathrel{|\!\sim}$ which satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Or we define its associated sv-model $W = \langle S, \ell, \prec \rangle$ by:

- $S = \{\langle m, \alpha \rangle \mid m \in \mathcal{U}, \alpha \in \mathcal{L}\}$,
- $\ell(\langle m, \alpha \rangle) = m$, and
- $\langle n, \alpha \rangle \prec \langle m, \beta \rangle$ iff $\begin{cases} m \models \alpha, \\ m \text{ is not normal for } \alpha \text{ with respect to } \mathrel{|\!\sim}, \\ n \models \alpha \text{ and} \\ \text{if } m \text{ is normal for } \beta \text{ with respect to } \mathrel{|\!\sim} \text{ then } n \not\models \beta. \end{cases}$

The intuition behind this definition is the same as in Definition 9.81. We want to show that the consequence relation based on this model is identical to the one we started with:

Lemma 9.89 (Completeness) For a consequence relation $\mathrel{|\!\sim}$ which satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Or, its associated sv-model W induces a consequence relation $\mathrel{|\!\sim}_W$ with $\mathrel{|\!\sim} = \mathrel{|\!\sim}_W$.

Proof: We want to show that $\ell[P_W(\alpha)] = \{m \in \mathcal{U} \mid m \text{ is normal for } \alpha \text{ with respect to } \mathrel{|\!\sim}\}$. Suppose we have $\langle m, \gamma \rangle$ with $m \models \alpha$ and m is not normal for α. We distinguish two cases:

- m is not normal for γ, then we have $\langle m, \alpha \rangle \prec \langle m, \gamma \rangle$, so $\langle m, \gamma \rangle \notin P_W(\alpha)$.

- m is normal for γ. Now we want to find a world n such that $n \models \alpha$ but $n \not\models \gamma$. So suppose $\models \alpha \to \gamma$. Then as m is not normal for α there exists a β such that $\alpha \mathrel{|\!\sim} \beta$ and $m \not\models \beta$. Then with $\models \alpha \to \gamma$ we have $\models \alpha \leftrightarrow (\gamma \wedge \alpha)$ and with Left Logical Equivalence we have $\gamma \wedge \alpha \mathrel{|\!\sim} \beta$ and with Conditionalization (which follows from the five rules) we have $\gamma \mathrel{|\!\sim} \alpha \to \beta$. However, since m is normal for γ we have $m \models \alpha \to \beta$ and $m \models \alpha$ so $m \models \beta$, in contradiction with our assumption. Therefore we have $\not\models \alpha \to \gamma$, so there exists an n with $n \models \alpha$ and $n \not\models \gamma$. Then we have $\langle n, \alpha \rangle \prec \langle m, \gamma \rangle$ and therefore $\langle m, \gamma \rangle \notin P_W(\alpha)$.

So if $m \in \ell[P_W(\alpha)]$ then m is normal for α. Now suppose m is normal for α. Then $\ell(\langle m, \alpha \rangle) \models \alpha$ (using Reflexivity). Furthermore, if we have $\langle n, \gamma \rangle \prec \langle m, \alpha \rangle$, then as m is normal for α we must have $n \not\models \alpha$, so $\langle m, \alpha \rangle \in P_W(\alpha)$, so $m \in \ell[P_W(\alpha)]$.
So then we get: $\alpha \mathrel{|\!\sim}_W \beta$ iff $\ell[P_W(\alpha)] \models \beta$ iff $\{m \in \mathcal{U} \mid m$ is normal for α with respect to $\mathrel{|\!\sim}\} \models \beta$ iff $\alpha \mathrel{|\!\sim} \beta$. Indeed we have $\mathrel{|\!\sim} = \mathrel{|\!\sim}_W$ as required. □

Theorem 9.90 (Representation for sv-models) A consequence relation $\mathrel{|\!\sim}$ satisfies Reflexivity, Left Logical Equivalence, And, Right Weakening, and Or if and only if $\mathrel{|\!\sim} = \mathrel{|\!\sim}_W$ for some sv-model W.

Proof: Follows immediately from Lemma 9.87 and Lemma 9.89. □

In the next subsection we will add the rule Weak Rational Monotonicity.

9.3.7 Adding the rule "Weak Rational Monotonicity"

It will turn out that the following property of models will do the trick for a representation result for the Rules 1-4, 6 and Weak Rational Monotonicity:

Definition 9.91 (Homogeneity of minimal states)

- An sv-model W satisfies *homogeneity of minimal states* if the following holds for all formulae α: if there is a minimal state in W satisfying α, then all α-minimal states are minimal.

- An *hms-model* is an sv-model satisfying homogeneity of minimal states. Its associated consequence relation is defined as in Definition 9.86.

First we will show that all hms-models satisfy all of the rules mentioned before, and Weak Rational Monotonicity.

Proposition 9.92 (Soundness) For any hms-model W, its associated consequence relation satisfies Reflexivity, Left Logical Equivalence, Right Weakening, And, Or and Weak Rational Monotonicity.

Proof: Since W is an sv-model, $\mid\!\sim_W$ satisfies rules 1 through 6. Now suppose $\not\mid\!\sim_W \neg\alpha$ and $\mid\!\sim_W \alpha \to \beta$. We have to prove that $\alpha \mid\!\sim_W \beta$. From $\not\mid\!\sim_W \neg\alpha$ we know that there exists a ($\top$-)minimal state which satisfies α. Let s be any α-minimal state, then because W satisfies homogeneity of minimal states, s is a minimal state. Therefore $\ell(s) \models \alpha \to \beta$ and since $\ell(s) \models \alpha$, we have that $\ell(s) \models \beta$. We conclude that $\alpha \mid\!\sim_W \beta$. □

To get a representation result, we need to find, given a consequence relation $\mid\!\sim$ which satisfies all six rules (Rules 1-4, 6 and 7), an hms-model W such that $\mid\!\sim_W = \mid\!\sim$. It turns out that the construction in Definition 9.88 will do this. In order to prove this, we first need the following lemma:

Lemma 9.93 Let $\mid\!\sim$ be a consequence relation which satisfies Reflexivity, Left Logical Equivalence, Right Weakening, And, Or and Weak Rational Monotonicity. Let W be the associated sv-model as defined in Definition 9.88. Then we have the following: If $\langle m, \beta\rangle$ is minimal, then for any formula φ, the state $\langle m, \varphi\rangle$ is also minimal.

Proof: Let $\mid\!\sim$ and W be as above and suppose $\langle m, \beta\rangle$ is minimal. Now suppose $\langle m, \varphi\rangle$ is *not* minimal. Then there exists a state $\langle n, \gamma\rangle$ with $\langle n, \gamma\rangle \prec \langle m, \varphi\rangle$. By definition we have that $n \models \gamma$, that $m \models \gamma$, that m is not normal for γ (with respect to $\mid\!\sim$), and that if m is normal for φ then $n \not\models \varphi$. We will distinguish two cases to prove that $\langle m, \beta\rangle$ is not minimal.

- Suppose m is not normal for β. Then it is easy to verify that $\langle n, \gamma\rangle \prec \langle m, \beta\rangle$.
- Suppose m is normal for β. Suppose that $\models \gamma \to \beta$, then $\models \gamma \leftrightarrow (\beta \wedge \gamma)$. As m is not normal for γ, there exists a formula δ such that $\gamma \mid\!\sim \delta$ but $m \not\models \delta$. With Left Logical Equivalence we have $\beta \wedge \gamma \mid\!\sim \delta$ and with Conditionalization (which is a derived rule) we have $\beta \mid\!\sim \gamma \to \delta$. As m was normal for β, we have $m \models \gamma \to \delta$ and because $m \models \gamma$ we have $m \models \delta$, a contradiction. So $\not\models \gamma \to \beta$ and we can find k with $k \models \gamma$ but $k \not\models \beta$. Then we have that $\langle k, \gamma\rangle \prec \langle m, \beta\rangle$.

In both cases we find that $\langle m, \beta\rangle$ is not minimal, a contradiction, so $\langle m, \varphi\rangle$ must be minimal. □

We are now ready to prove that the construction of Definition 9.88 can be used again.

Lemma 9.94 For a consequence relation $\mid\!\sim$ which satisfies Reflexivity, Left Logical Equivalence, Right Weakening, And, Or, and Weak Rational Monotonicity, its associated sv-model W satisfies homogeneity of minimal states.

Proof: Suppose $\langle m, \beta\rangle$ is minimal and $m \models \alpha$. Suppose $\langle n, \gamma\rangle$ is α-minimal but not minimal. From the contraposition of Lemma 9.93 we know that for any formula φ, the state $\langle n, \varphi \rangle$ is not minimal. In the proof of Lemma 9.89 we saw that $\ell[P_W(\top)] = \{m \in \mathcal{U} \mid m$ is normal for $\top$ with respect to $\mid\!\sim\}$. As n is not in $\ell[P(\top)]$ we have that n is not normal for $\top$. Distinguishing two cases, we will show that $\langle n, \gamma \rangle$ is not α-minimal:

- Suppose that n is not normal for γ, then $\langle m, \top \rangle \prec \langle n, \gamma \rangle$ and $m \models \alpha$.

- Suppose that n is normal for γ. Suppose that $\models \alpha \rightarrow \gamma$. As n is not normal for $\top$ there exists a formula δ such that $\mid\!\sim \delta$ but $n \not\models \delta$. As $\mid\!\sim \delta$ and $\models \delta \rightarrow (\alpha \rightarrow \delta)$ we have with Right Weakening that $\mid\!\sim \alpha \rightarrow \delta$. Because m is normal for $\top$ (again because $\ell[P(\top)] = \{m \in \mathcal{U} \mid m$ is normal for $\top$ with respect to $\mid\!\sim\}$), and $m \not\models \neg\alpha$, we have that $\not\mid\!\sim \neg\alpha$. With Weak Rational Monotonicity we conclude that $\alpha \mid\!\sim \delta$. Furthermore, since $\models \alpha \rightarrow \gamma$, we have $\models \alpha \leftrightarrow (\gamma \wedge \alpha)$, so with Left Logical Equivalence we have $\gamma \wedge \alpha \mid\!\sim \delta$. Conditionalization yields $\gamma \mid\!\sim \alpha \rightarrow \delta$. As n is normal for γ, we have $n \models \alpha \rightarrow \delta$. Also $n \models \alpha$ ($\langle n, \gamma \rangle$ is α-minimal), so $n \models \delta$ which is a contradiction. Therefore $\not\models \alpha \rightarrow \gamma$, so there is a k with $k \models \alpha$ and $k \not\models \gamma$. But then we have that $\langle k, \top \rangle \prec \langle n, \gamma \rangle$ and $k \models \alpha$.

In both cases we conclude that $\langle n, \gamma \rangle$ is not α-minimal, which is a contradiction. Therefore $\langle n, \gamma \rangle$ is minimal. $\square$

Now we can prove the representation result.

Theorem 9.95 (Representation for hms-models) A consequence relation $\mid\!\sim$ satisfies Reflexivity, Left Logical Equivalence, Right Weakening, And, Or and Weak Rational Monotonicity if and only if $\mid\!\sim \, = \, \mid\!\sim_W$ for some hms-model W.

Proof: The direction from right to left is given in Proposition 9.92. For the other direction we use Theorem 9.90 to find an sv-model W such that $\mid\!\sim \, = \, \mid\!\sim_W$. Lemma 9.94 ensures that W is an hms- model. $\square$

9.3.8 Other rules

A number of rules which are mentioned in [KLM90] have not been treated yet. Naturally, we have not included CM, which we reject. A rule which we reject for similar reasons, is Consistency Preservation (CP):

$$\text{If } \alpha \mid\!\sim \bot \text{ then } \alpha \models \bot \qquad (\textit{Consistency Preservation})$$

The restriction on our models corresponding to this rule is very easy: every consistent formula should have a minimal state satisfying it. This is similar to, but independent of smoothness: there are smooth models in which CP is not satisfied, and there are non-smooth models where it is satisfied. But we reject this condition for a similar reason: we do allow models in which for every state satisfying a certain formula, there is a more preferred state in which it is also true.

Another restricted variant of monotonicity is the following rule:

If $\alpha \mid\!\sim \gamma$ and $\alpha \not\mid\!\sim \neg\beta$ then $\alpha \wedge \beta \mid\!\sim \gamma$ (*Rational Monotonicity*)

Since this rule allows monotonicity under (intuitively) weaker circumstances than CM, we have not incorporated it.

When rejecting CM, we even *have to* reject at least one of Rational Monotonicity and Consistency Preservation:

Proposition 9.96 Consistency Preservation and Rational Monotonicity imply Cautious Monotonicity in the presence of And, Reflexivity, and Right Weakening.

Proof: Suppose $\alpha \mid\!\sim \beta$ and $\alpha \mid\!\sim \gamma$.

- If $\alpha \mid\!\sim \neg\beta$ then $\alpha \mid\!\sim \beta \wedge \neg\beta$ (And), so that CP gives $\alpha \models \beta \wedge \neg\beta$, but then $\alpha \wedge \beta \models \bot$ and $\alpha \wedge \beta \models \gamma$. Using Reflexivity and Right Weakening we conclude $\alpha \wedge \beta \mid\!\sim \gamma$.
- If $\alpha \not\mid\!\sim \neg\beta$ then with RM we have $\alpha \wedge \beta \mid\!\sim \gamma$.

□

An alternative rule (which has not been mentioned in the literature, to the best of our knowledge) that together with Rational Monotonicity also implies CM, is Inconsistency Monotonicity:

If $\alpha \mid\!\sim \bot$ then $\alpha \wedge \beta \mid\!\sim \bot$ (*Inconsistency Monotonicity*)

In the presence of Right Weakening, this is implied by CM. The same argument as before holds for our rejection of this rule.

We also mention the following two rules:

If $\alpha \mid\!\sim \beta$ and $\beta \models \alpha$ then $(\alpha \mid\!\sim \gamma \Leftrightarrow \beta \mid\!\sim \gamma)$ (*Cumulativity*)

If $\alpha \mid\!\sim \beta$ and $\beta \mid\!\sim \alpha$ then $(\alpha \mid\!\sim \gamma \Leftrightarrow \beta \mid\!\sim \gamma)$ (*Reciprocity*)

The latter rule is also called Equivalence. Both of these rules imply CM:

Proposition 9.97 Cumulativity implies Cautious Monotonicity in the presence of Reflexivity and And. Moreover, Reciprocity implies Cautious Monotonicity in the presence of And, Reflexivity, and Right Weakening.

Proof:

- Suppose $\alpha \mid\!\sim \beta$ and $\alpha \mid\!\sim \gamma$. Using Reflexivity we have $\alpha \mid\!\sim \alpha$ and And gives $\alpha \mid\!\sim \alpha \wedge \beta$. Furthermore, $\alpha \wedge \beta \models \alpha$, so Cumulativity gives $\alpha \wedge \beta \mid\!\sim \gamma$.

- Suppose $\alpha \mid\!\sim \beta$ and $\alpha \mid\!\sim \gamma$. With Reflexivity and And we have $\alpha \mid\!\sim \alpha \wedge \beta$. As $\alpha \wedge \beta \models \alpha$, using Reflexivity and Right Weakening, we have $\alpha \wedge \beta \mid\!\sim \alpha$. Reciprocity implies that $\alpha \wedge \beta \mid\!\sim \gamma$.

□

As these two rules are stronger than CM, we do not treat them.

9.3.9 Conclusions and related work

We have given a classification of inference relations that do not (necessarily) satisfy Cautious Monotonicity. Representation results were given for a number of classes of such inference relations, starting from a class larger than the class of cumulative inference relations as defined in [KLM90]. Given the fact that known nonmonotonic formalisms as default logic and circumscription, but also our own MTEL, have non-cumulative inference relations, it seems worthwhile to investigate properties of such relations. We do not share the opinion of [Gab85] and [KLM90] that such relations can not be considered to be "logical systems". They only give a weak (intuitive) argumentation for this statement, and systems satisfying (just) the basic four rules (1-4) would also seem to be logical systems, so maybe the border between logical and non-logical should be drawn there. The current section can be seen as supplementing the work in [KLM90] to the non-cumulative case.

All of the rules described in this section can be disputed, even the basic four rules (of those four, maybe especially the And rule). These four rules are necessary if one wants to give semantics based on picking out the preferred (classical) interpretations, to inference relations. Of course many different combinations of rules can be considered. The selection made in this section is inspired by the rules in [KLM90] and the rules satisfied by MTEL.

One rule we have not considered here is Loop [KLM90], corresponding to transitivity of the preference relation for cumulative relations. Transitivity seems a plausible property for any preference relation, and we would like to find a rule which characterizes it in the non-cumulative case; it may be Loop, but this is not necessarily the case for non-cumulative relations. Other rules from [KLM90] we have not considered are Disjunctive Rationality and Negation Rationality.

Apart from [KLM90], there are a number of other publications dealing with various kinds of preferential models and properties of their associated consequence relation. It started with [Sho87], and was continued by [Mak89]. In [Voo93], a representation result is given in terms of inference operations (see Section 10.1) and so-called *epistemic preference models*, which are just like our ordered models, with the set of states equal to the set of closed S5-models (see Definition 4.8), and the labeling function is the identity function. The required properties of the inference operations are inclusion (corresponding to Reflexivity), cumulative transitivity (corresponding to Cut) and $\mathcal{L}$-invariance (corresponding to Left logical Equivalence) (And is not incorporated here since the language and the models for it are kept abstract; there may not even be conjunction in the language). This would suggest that the labeling function of an ordered model is not necessary. More representation results for the infinitary case (inference operations) can be found in [Sch92b, Sch95a].

Acknowledgments

The results of Section 9.1 are contained in [Eng96a]. The material in Section 9.2 appeared earlier in [Eng98]. The material of the last section is contained in [Eng].

Chapter 10

Belief Set Operators

In Chapter 1 we introduced four levels of abstraction for describing reasoning processes. A formalization of descriptions on the first three levels was given in Chapter 2. On level two, reasoning can be described by multiple belief state operators (MBSOs). These operators assign a set of information states to each set of input formulae. In this chapter we will concentrate on MBSOs of the following information state frame. The language $\mathcal{L}$ is a propositional language, with the consequence operator Cn, and the states are sets of formulae (not necessarily closed under Cn). The (natural) information ordering is set-inclusion. (Note that this information state frame is almost the same as the frame of syntactical states IS^{syn}, but without the restriction of states to sets closed under provability.) An MBSO in this information state frame will be called a *belief set operator*, and these operators are almost the same as the multi-interpretation operators of Section 8.2, with two differences: a multi-interpretation operator may have a different input and output language, whereas these are identical for belief set operators; the other difference is that the family of sets of formulae assigned by a multi-interpretation operator must consist of sets closed under provability, whereas this is not required for belief set operators. In contrast to the rest of this thesis, in this chapter we will call *any* set of formulae, not necessarily closed under provability, a belief set.

FBSOs of the same information state frame as above will also be introduced, and will be called *inference operations*. In this chapter we will study belief set operators and selection functions, the link with inference operations, and the link with semantical operators.

10.1 Inference operations

The study of inference operations is already well-established (see [Mak94] for an overview). In this section, we will briefly review the main notions, properties and results concerning inference operations. For completeness, let us give a formal definition.

Definition 10.1 (Inference operation) Let $\mathcal{L}$ be a language. An *inference operation* is a function $C : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathcal{L})$. The pair $(\mathcal{L}, C)$ is called an *inference system.*

Inference operations can be formalizations of nonmonotonic reasoning processes, but also of more classical closure operations, modeling classical logics.

Definition 10.2 (Closure operation) An inference system $(\mathcal{L}, C_{\mathcal{L}})$ is a *closure system* (and $C_{\mathcal{L}}$ a *closure operation*) if it satisfies the following conditions:

1. $X \subseteq C_{\mathcal{L}}(X)$ (*inclusion*),
2. $C_{\mathcal{L}}(C_{\mathcal{L}}(X)) = C_{\mathcal{L}}(X)$ (*idempotence*), and
3. $X \subseteq Y \Rightarrow C_{\mathcal{L}}(X) \subseteq C_{\mathcal{L}}(Y)$ (*monotony*).

Inference operations that formalize a classical compact kind of logic are called deductive:

Definition 10.3 (Deductive inference operation) A closure system $(\mathcal{L}, C_{\mathcal{L}})$ is a *deductive system* (and $C_{\mathcal{L}}$ is a *deductive (inference) operation*), if $C_{\mathcal{L}}$ satisfies *compactness*: whenever $\varphi \in C_{\mathcal{L}}(X)$, there must exist a finite subset $Y \subseteq X$ such that $\varphi \in C_{\mathcal{L}}(Y)$.

Let $(\mathcal{L}_0, Cn)$ denote the inference system based on classical propositional logic. A semantics for a closure system $(\mathcal{L}, C_{\mathcal{L}})$ can be defined by a model-theoretic system.

Definition 10.4 (Model-theoretic system) A model-theoretic system $(\mathcal{L}, \mathsf{Mod}, \models)$ is determined by a language $\mathcal{L}$, a set (or class) Mod whose elements are called *worlds* and a *relation of satisfaction* $\models \subseteq \mathsf{Mod} \times \mathcal{L}$ between worlds and formulae. Given such a system we define:

1. For $X \subseteq \mathcal{L}$, define $Mod^{\models}(X) = \{m : m \in \mathsf{Mod} \text{ and } m \models X\}$, where $m \models X$ if for every $\varphi \in X : m \models \varphi$.
2. For $K \subseteq \mathsf{Mod}$, define $Th^{\models}(K) = \{\varphi : \varphi \in \mathcal{L} \text{ and } K \models \varphi\}$, where $K \models \varphi$ if for all $m \in K : m \models \varphi$.
3. For $X \subseteq \mathcal{L}$, define $C^{\models}(X) = \{\varphi : Mod^{\models}(X) \subseteq Mod^{\models}(\varphi)\}$, and $X \models \varphi$ if $\varphi \in C^{\models}(X)$.

It is not difficult to verify that $(\mathcal{L}, C^{\models})$ is a closure system and if $C^{\models}(X) = X$ then $Th^{\models}(Mod^{\models}(X)) = X$. A system $(\mathcal{L}, \mathsf{Mod}, \models)$ is called *compact* if the closure operation $C^{\models}$ is compact.

A model-theoretic system is meant to give semantics to an inference operation.

Definition 10.5 An inference system is $(\mathcal{L}, C_{\mathcal{L}})$ *correct* with respect to a model-theoretic system $(\mathcal{L}, \mathsf{Mod}, \models)$ if $C_{\mathcal{L}}(X) \subseteq C^{\models}(X)$, and *complete* if $C_{\mathcal{L}}(X) = C^{\models}(X)$.

So deductive inference operations formalize 'classical' logics. Nonmonotonic reasoning is a method of inferring extra conclusions, 'on top of' what the deductive operation can yield. This nonmonotonic reasoning can again be formalized by an inference operation, but this time one not satisfying monotony.

A condition on inference operations is said to be *pure* if it concerns the operation alone without regard to its interrelations to a deductive system $(\mathcal{L}, C_{\mathcal{L}})$ representing an underlying monotonic and compact logic. The most important pure conditions are the following (the first two were already mentioned in Section 2.1 for FBSOs in general; also they were mentioned as conditions on consequence relations in Section 9.3):

Definition 10.6 Let C be an inference operation. We define the following pure properties of C.

- $X \subseteq Y \subseteq C(X) \Rightarrow C(Y) \subseteq C(X)$ (*cut*),
- $X \subseteq Y \subseteq C(X) \Rightarrow C(X) \subseteq C(Y)$ (*cautious monotony*),
- $X \subseteq Y \subseteq C(X) \Rightarrow C(X) = C(Y)$ (*cumulativity*).

Some impure conditions are

- $C(X) \cap C(Y) \subseteq C(C_{\mathcal{L}}(X) \bigcap C_{\mathcal{L}}(Y))$ (*distributivity*),
- $C_{\mathcal{L}}(X) \neq L \Rightarrow C(X) \neq L$ (*consistency preservation*).

Definition 10.7 (Inference frame)

1. A system $\mathcal{IF} = (\mathcal{L}, C_{\mathcal{L}}, C)$ is said to be an *inference frame* if $C_{\mathcal{L}}$ is a deductive inference operation on $\mathcal{L}$, and C is an inference operation satisfying :
   $C_{\mathcal{L}}(X) \subseteq C(X)$ for all $X \subseteq \mathcal{L}$ (*supradeductivity*).
2. The operation C satisfies *left absorption* if $C_{\mathcal{L}}(C(X)) = C(X)$, and C satisfies *congruence* or *right absorption* if $C_{\mathcal{L}}(X) = C_{\mathcal{L}}(Y) \Rightarrow C(X) = C(Y)$. C satisfies *full absorption* if C satisfies left absorption and congruence.
3. An inference frame $\mathcal{DF} = (\mathcal{L}, C_{\mathcal{L}}, C)$ is said to be a *deductive inference frame* if it satisfies full absorption.

If $C_{\mathcal{L}} = Cn$ then the property of supradeductivity is usually called *supraclassicality*.

The semantics of a deductive frame can be described by introducing a model operator based on a model-theoretic system ([DH94]). In fact, such a model operator can again be seen as an FBSO, but this time based on the information state frame of epistemic states, IS^{ep}.

Definition 10.8 (Semantical frame)

1. $\mathcal{SF} = (\mathcal{L}, \mathsf{Mod}, \models, \Phi)$ is a *semantical frame* if $(\mathcal{L}, \mathsf{Mod}, \models)$ is a model-theoretic system and $\Phi : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathsf{Mod})$ is a functor (called *model operator*) such that $\Phi(X) \subseteq Mod^{\models}(X)$.

2. Let $C_\Phi(X) = Th^{\models}(\Phi(X))$. The operator Φ is said to be $C_{\mathcal{L}}$-*invariant* if $(\forall X \subseteq \mathcal{L})(\Phi(X) = \Phi(C_{\mathcal{L}}(X)))$.

Note that the inference operation C_Φ satisfies supradeductivity, and hence $(\mathcal{L}, C^{\models}, C_\Phi)$ is an inference frame associated to $\mathcal{SF}$. Again we can define completeness of a syntactical notion with respect to a corresponding semantics.

Definition 10.9 An inference frame $\mathcal{I} = (\mathcal{L}, C_{\mathcal{L}}, C)$ is said to be *complete* for a semantical frame $(\mathcal{L}, \mathsf{Mod}, \models, \Phi)$ if $(\mathcal{L}, C_{\mathcal{L}})$ is complete with respect to $(\mathcal{L}, \mathsf{Mod}, \models)$ and $C = C_\Phi$.

Representation theorems for classes of inference frames can be proved by using semantical frames based on the *Lindenbaum-Tarski* construction of maximal consistent sets. We recall the ingredients of this construction. Let $(\mathcal{L}, C_{\mathcal{L}})$ be a deductive system. A set $X \subseteq \mathcal{L}$ is said to be relatively maximal consistent (*r-maximal*) iff there is a formula $\varphi \in \mathcal{L}$ such that $\varphi \notin C_{\mathcal{L}}(X)$ and for every proper super set $Y \supset X$ the condition $\varphi \in C_{\mathcal{L}}(Y)$ is satisfied. Let $rmax(\mathcal{L})$ be the set of all r-maximal subsets of $\mathcal{L}$. The Lindenbaum-Tarski semantics (abbreviated by LT-semantics) is defined by the model-theoretic system $(\mathcal{L}, M, \models)$ where $M = rmax(\mathcal{L})$ and $m \models \varphi$ iff $\varphi \in m$. Then $C^{\models} = C_{\mathcal{L}}$. We collect some elementary results that can be formulated and proved within this framework (see [DH94]).

Proposition 10.10 Let $\mathcal{F} = (\mathcal{L}, C_{\mathcal{L}}, C)$ be an inference frame satisfying left absorption. Then there exists a semantical frame $\mathcal{SF} = (\mathcal{L}, M, \models, \Phi)$ such that $\mathcal{F}$ is complete with respect to $\mathcal{SF}$, i.e. $C_{\mathcal{L}} = C^{\models}$ and $C = C_\Phi$.

Proof: Let $(\mathcal{L}, M, \models)$ be the LT-semantics for $(\mathcal{L}, C_{\mathcal{L}})$ and $\Phi(X) = \{m : m \in M, C(X) \subseteq m\}$. It is easy to show that $C = C_\Phi$. $\square$

Left absorption does not imply congruence. We get a completeness result for deductive inference frames by using invariant semantical frames ([DH94]).

Proposition 10.11

1. Let $\mathcal{F} = (\mathcal{L}, C_{\mathcal{L}}, C)$ be a deductive inference frame. Then there exists a semantical frame $\mathcal{S} = (\mathcal{L}, M, \models, \Phi)$ such that Φ is an invariant model operator and $\mathcal{F}$ is complete for $\mathcal{S}$.

2. If Φ is an invariant model operator for the logical system $(\mathcal{L}, M, \models)$ then $(\mathcal{L}, C^{\models}, C_\Phi)$ is a deductive inference frame.

Proof:

1. Let $(\mathcal{L}, M, \models)$ be the LT-semantics for $(\mathcal{L}, C_\mathcal{L})$ and $\Phi(X) = Mod^{\models}(C(X))$. Left-absorption implies $C_\Phi = C$. Invariance of Φ follows from right absorption: since $C(C_\mathcal{L}(X)) = C(X)$ we have $\Phi(X) = Mod^{\models}(C(X)) = Mod^{\models}(C(C_\mathcal{L}(X)))$ $= \Phi(C_\mathcal{L}(X))$.
2. Let $(\mathcal{L}, C^{\models}, C_\Phi)$ be a semantical frame and Φ an invariant model operator. By definition $C_\Phi(X) = Th^{\models}(\Phi(X))$. Hence $C_\mathcal{L}(C_\Phi(X)) = C_\Phi(X)$. By invariance of Φ we have $\Phi(X) = \Phi(C_\mathcal{L}(X))$, hence $C_\Phi(X) = C_\Phi(C_\mathcal{L}(X))$, i.e. C_Φ satisfies right-absorption.

$\square$

10.2 Properties of belief set operators

A belief set operator was defined before as an MBSO in the information state frame of sets of formulae; the FBSO associated to a belief set operator is then fixed by Definition 2.17. Such an associated FBSO (or inference operation) will be called the *kernel* of the belief set operator. To make these notions more clear, we will give a direct definition. Remember that in this chapter, a belief set is any set of formulae.

Definition 10.12 (Belief set operator) A belief set operator B is a function that assigns a belief set family to each set of initial facts: $B : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathcal{P}(\mathcal{L}))$. The *kernel* of B, denoted K_B, is defined by $K_B(X) = \bigcap B(X)$.

The properties of inclusion, non-inclusiveness and invariance are as defined in Definition 2.15. Examples of belief set operators (for default logic, autoepistemic logic and belief revision) are given in that same section; we will give one extra example here.

Example 10.13 (Poole Systems) Let $\Sigma = (D, E)$, $D \cup E \subseteq \mathcal{L}$; the elements of D are called defaults, the elements of E are said to be constraints. A set $\delta \subseteq D$ is a basis for $X \subseteq \mathcal{L}$ if the set $X \cup \delta \cup E$ is consistent and δ is maximal with this property. Let $Cons_\Sigma(X) = \{\delta : \delta \subseteq D$ and δ is a basis for $X\}$. Then define $B_\Sigma(X) = \{Cn(X \cup \delta) : \delta \in Cons_\Sigma(X)\}$. Then B_Σ is a belief set operator.

Structural properties of inference operations (like monotony, cut or cautious monotony) can be generalized to properties of belief set operators, usually in more than one way. The simplest way is to relate everything to the kernel. For instance, we could say that B is monotonic if and only if its kernel K_B is. But this definition does not at all consider the structure of the belief set families, and we can define more refined versions of such properties that *do* take into account this structure.

In order to define these properties, it will be convenient to introduce an information ordering on belief set families. For belief sets there is already a natural notion of degree of information (a belief set T contains more information than a belief set S if $S \subseteq T$). Using this new ordering of information, the properties of belief set operators resemble their counterparts for inference operations.

Definition 10.14 Let $\mathcal{A}$, $\mathcal{B}$ be belief set families. We say $\mathcal{B}$ contains more information than $\mathcal{A}$, denoted $\mathcal{A} \preceq \mathcal{B}$, if $(\forall T \in \mathcal{B})(\exists S \in \mathcal{A})(S \subseteq T)$. We write $\mathcal{A} \equiv \mathcal{B}$ if $\mathcal{A} \preceq \mathcal{B}$ and $\mathcal{B} \preceq \mathcal{A}$.

If one of the arguments in the above definition is a singleton belief set family, we will often omit the parentheses and write $X \preceq \mathcal{A}$ instead of $\{X\} \preceq \mathcal{A}$. Thus, we can also write $X \preceq Y$ instead of $X \subseteq Y$. In words this definition says that a belief set family $\mathcal{B}$ is considered to have more information than $\mathcal{A}$ if any of the sets of $\mathcal{B}$ extends some of the sets of $\mathcal{A}$. This also means that it may happen that a belief set in $\mathcal{A}$ has no extending belief set in $\mathcal{B}$. One can think of the belief sets as (partial) possible worlds: the less possible worlds the agent considers, the more sure she is of the state of affairs of the outside world. So the more possibilities, the less knowledge an agent has. On the other hand, the possible states in $\mathcal{B}$ must contain more information (or the same information) than their counterparts in $\mathcal{A}$. Note that this condition implies that $\bigcap \mathcal{A} \subseteq \bigcap \mathcal{B}$. We introduce the following formal properties of belief set operators capturing essential features of a rational agent.

Definition 10.15 Let B be a belief set operator.

1. B satisfies *belief monotony* if $(\forall X \forall Y)(X \preceq Y \Rightarrow B(X) \preceq B(Y))$.
2. B satisfies *weak belief monotony* if $(\forall XY)(X \preceq Y \preceq B(X) \Rightarrow B(X) \preceq B(Y))$.
3. B satisfies *belief transitivity* if $(\forall XYS)(S \in B(X)$ and $X \subseteq Y \subseteq S \Rightarrow K_B(Y) \subseteq S)$.[1]
4. B satisfies *belief cut* if $(\forall XY)(X \preceq Y \preceq B(X) \Rightarrow B(Y) \preceq B(X))$.
5. B satisfies *belief cumulativity* if it satisfies weak belief monotony and belief cut.
6. B satisfies *strong belief cumulativity* if it satisfies belief cumulativity and belief transitivity.
7. B satisfies *strong belief cut* if $(\forall XYS)(S \in B(X)$ *and* $X \subseteq Y \subseteq S \Rightarrow (\exists T \in B(Y))(T \subseteq S))$.

[1]This property is called *cumulative transitivity* in [Voo93].

It is easy to check that strong belief cut implies belief cut and belief transitivity.

In [Voo93] a belief set operator B satisfying inclusion is said to be cumulative if it satisfies belief transitivity and the following condition: $(\forall XYS)(S \in B(X)$ and $X \subseteq Y \subseteq S \Rightarrow B(Y) \subseteq B(X))$. A weaker form of this notion is defined by the following condition: $(\forall XYS)(S \in B(X)$ and $X \subseteq Y \subseteq S \Rightarrow B(X) \preceq B(Y))$. All these properties are generalizations of the notion of cautious monotony for inference operations to the case of belief set operators. Similarly, there are alternative versions of the generalization of cut and cumulativity to belief set operators. There is not yet a complete analysis of these properties and their interrelations. The following holds:

Proposition 10.16 Let B be a belief set operator satisfying inclusion.

1. If B is belief monotonic then K_B is monotonic.
2. If B satisfies belief transitivity or belief cut then K_B satisfies cut.
3. If B satisfies weak belief monotony then K_B satisfies cautious monotony.

Proof:

1. $X \subseteq Y \Rightarrow B(X) \preceq B(Y) \Rightarrow \bigcap B(X) \subseteq \bigcap B(Y)$.
2. Suppose B satisfies belief cut, and suppose $X \subseteq Y \subseteq K_B(X)$, then certainly $X \preceq Y \preceq B(X)$, so $B(Y) \preceq B(Y)$ whence $\bigcap B(Y) \preceq \bigcap B(Y)$. Now suppose B satisfies belief transitivity, and suppose $X \subseteq Y \subseteq K_B(X)$. Let $T \in B(X)$, then $X \subseteq Y \subseteq T$ so $\bigcap B(Y) \subseteq T$. It follows that $\bigcap B(Y) \subseteq \bigcap B(X)$.
3. If $X \subseteq Y \subseteq \bigcap B(X)$ then $X \preceq Y \preceq B(X)$ so $B(X) \preceq B(Y)$. It follows that $\bigcap B(X) \subseteq \bigcap B(Y)$.

□

So all of the properties of Definition 10.15 are generalizations of the corresponding properties of inference operations. Given a belief set operator B with desirable properties, the associated inference operation K_B has analogous properties.

Given an inference operation C, there are of course in general many belief set operators B such that $K_B = C$, the most trivial being $B(X) = \{C(X)\}$. One could ask whether there are *non-trivial* belief set operators B with $K_B = C$ which have interesting structural properties, and if there is a general way of obtaining them. The results in [Mak94], building on results in [KLM90], show that this can be done using preferential models. We will briefly sketch this. A preferential model is a triple $\langle \mathsf{Mod}, \models, < \rangle$ where Mod is a set of states, $\models$ is any relation between states and formulae and $<$ is a relation between models. A state $m \in \mathsf{Mod}$ *preferentially satisfies* a set of formulae A, denoted $m \models_< A$, if $m \models A$ and there is no $n \in \mathsf{Mod}$ such that $n < m$ and $n \models A$. An inference operation $C_<$ can then be defined by

$C_<(X) = \{\varphi \in \mathcal{L} \mid \forall m \in \mathsf{Mod}, m \models_< X \Rightarrow m \models \varphi\}$. A preferential model is called *smooth*, if for any $X \subseteq \mathcal{L}$ and $m \in \mathsf{Mod}$ such that $m \models X$, there exists a state $n \in \mathsf{Mod}$ such that $n \leq m$ and $n \models_< X$ (cf. Section 9.3). The basic result of [KLM90], proved independently by [Mak89], is that for any cumulative inference operation C, there is a smooth preferential model $\langle \mathsf{Mod}, \models, < \rangle$ such that $C = C_<$. But this also gives rise to a belief set operator, in the sense that the theory of each state preferentially satisfying X can be seen as a belief set. If we set (in the notation of [Mak94]) $E_m = \{\varphi \in \mathcal{L} \mid m \models \varphi\}$ for each $m \in \mathsf{Mod}$, then a belief set operator B can be defined by $B(X) = \{E_m \mid m \models_< X\}$. It is easy to see that $K_B = C_<$. Moreover, this belief set operator satisfies the properties defined in Definition 10.15.

Proposition 10.17 Given a cumulative inference operation C, there exists a non-trivial belief set operator B satisfying all the properties in Definition 10.15 such that $K_B = C$.

Proof: Given C, let B be defined as above. Then B satisfies weak belief monotony: suppose $X \preceq Y \preceq B(X)$. Let $E_m \in B(Y)$, then $m \models_< Y$ so $m \models X$, and by smoothness there exists $n \leq m$ such that $n \models_< X$. As $Y \preceq B(X)$ we have $n \models Y$, so $n = m$. We have found $E_n \subseteq E_m$ and $E_n \in B(X)$ so $B(X) \preceq B(Y)$. Furthermore, B satisfies strong belief cut. Suppose $E_m \in B(X)$ and $X \subseteq Y \subseteq E_m$, then $m \models Y$ so there exists $n \leq m$ such that $n \models_< Y$. Since $X \subseteq Y$ we have $n \models X$ so $n = m$. We have found $E_n \in B(Y)$ such that $E_n \subseteq E_m$. These two properties imply all the other ones. $\square$

Propositions 10.16 and 10.17 state that when going from a level 2 description (belief set operators) to a level 1 description (inference operations), properties of well-behavedness transfer, and that we can go from a level 1 description to a level 2 description without losing such properties.

10.3 Belief frames

We now connect a belief set operator with a compact monotonic logic which can be considered as a deductive basis.

Definition 10.18

1. A system $\mathcal{BF} = (\mathcal{L}, C_{\mathcal{L}}, B)$ is said to be a *belief set frame* if the following conditions are satisfied:
   (a) $\mathcal{L}$ is a language and $C_{\mathcal{L}}$ is a *deductive inference operation* on $\mathcal{L}$.
   (b) B is a belief set operator on $\mathcal{L}$ satisfying non-inclusiveness and inclusion.
2. B satisfies *belief left absorption* iff $C_{\mathcal{L}}(T) = T$ for every $T \in B(X)$, and B satisfies *belief congruence* or *$C_{\mathcal{L}}$-invariance* iff $C_{\mathcal{L}}(X) = C_{\mathcal{L}}(Y)$ implies

$B(X) = B(Y)$. B satisfies *full absorption* iff B satisfies left belief absorption and congruence.

3. A belief set frame $\mathcal{DF} = (\mathcal{L}, C_{\mathcal{L}}, B)$ is said to be a *deductive belief set frame* if it satisfies full absorption.

Proposition 10.19 Let $\mathcal{BF} = (\mathcal{L}, C_{\mathcal{L}}, B)$ be a belief set frame satisfying strong belief cumulativity. Then $\mathcal{BF}$ satisfies belief left absorption and belief congruence, i.e. $\mathcal{BF}$ is a deductive belief set frame.

Proof:

- From belief transitivity it follows that for every $T \in B(X)$ the condition $K_B(T) \subseteq T$ is satisfied, hence $K_B(T) = T$. By supradeductivity we get $C_{\mathcal{L}}(T) \subseteq K_B(T)$, thus $C_{\mathcal{L}}(T) = T$.

- Assume $C_{\mathcal{L}}(X) = C_{\mathcal{L}}(Y)$. Since $K_B : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathcal{L})$ is cumulative, it follows that $(\mathcal{L}, C_{\mathcal{L}}, K_B)$ is a deductive frame, hence $K_B(X) = K_B(Y)$. It is sufficient to prove $B(X) = B(K_B(X))$, because this condition implies $B(X) = B(Y)$.

  Let $S \in B(X)$, by belief cut there is an extension $T \in B(K_B(X))$ such that $T \subseteq S$. By weak belief monotony there exists an $S_1 \in B(X)$ satisfying $S_1 \subseteq S$. Because the sets in $B(X)$ are pairwise non-inclusive we get $S = S_1$, which implies $T = S$, hence $S \in B(K_B(X))$.

  Let $T \in B(K_B(X))$; by weak belief monotony there is an $S \in B(X)$ such that $S \subseteq T$. By the previous proved condition this implies $S \in B(K_B(X))$, hence by non-inclusiveness of B we get $T = S$.

$\square$

Further important impure properties of inference frames can be generalized to belief set frames.

Definition 10.20 Let $(\mathcal{L}, C_{\mathcal{L}}, B)$ be a belief set frame.

1. B satisfies *belief distribution* if
   $(\forall XYS)((S \in B(C_{\mathcal{L}}(X) \bigcap C_{\mathcal{L}}(Y)) \Rightarrow (S \in B(X) \text{ or } S \in B(Y)))$.

2. B satisfies *belief consistency preservation* if
   $(\forall X)(C_{\mathcal{L}}(X) \neq \mathcal{L} \Rightarrow B(X) \neq \{\mathcal{L}\} \text{ and } B(X) \neq \emptyset)$.

In the last condition, both when $B(X) = \{\mathcal{L}\}$ and when $B(X) = \emptyset$, the input can be considered 'nonmonotonically inconsistent'. Both possibilities occur in for instance default logic: there are default theories with just one inconsistent extension, and there are default theories without extensions.

The following proposition holds.

Proposition 10.21

1. If B satisfies belief distribution then K_B satisfies distributivity.
2. If B satisfies belief consistency preservation then K_B satisfies consistency preservation.

Proof:

1. Suppose B satisfies belief distribution. Take any $S \in B(C_{\mathcal{L}}(X) \cap C_{\mathcal{L}}(Y)$, then $S \in B(X)$ or $S \in B(Y)$ so $\bigcap B(X) \subseteq S$ or $\bigcap B(Y) \subseteq S$. In both cases we have $\bigcap B(X) \cap \bigcap B(Y) \subseteq S$. It follows that $K_B(X) \cap K_B(Y) \subseteq \bigcap B(C_{\mathcal{L}}(X) \cap C_{\mathcal{L}}(Y)$.
2. Suppose $C_{\mathcal{L}}(X) \neq \mathcal{L}$, then $B(X) \neq \{\mathcal{L}\}$ and $B(X) \neq \emptyset$ from which we immediately get $\bigcap B(X) \neq \mathcal{L}$.

□

The semantics of a belief set is a set of models. Since there can be many belief sets we have to take into consideration functors associating to sets of assumptions sets of sets of models. Such functors are called *semantic belief state operators*. They are again MBSOs, but semantic ones, defined in the information state frame of epistemic states. We will nonetheless give a definition, including the definition of some properties defined for MBSOs in general.

Definition 10.22 (Semantic belief state operator)

1. A semantic belief state operator Γ is a function $\Gamma : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathcal{P}(\mathsf{Mod}))$.
2. The tuple $(\mathcal{L}, \mathsf{Mod}, \models, \Gamma)$ is said to be a *semantic belief state frame*.
3. Γ satisfies non-inclusiveness if $(\forall K J \in \Gamma(X))(J \subseteq K \Rightarrow J = K)$.
4. Γ satisfies inclusion if $(\forall X)(\forall K \in \Gamma(X))(K \subseteq Mod(X))$.
5. Γ satisfies left absorption, or L-invariance, if $\Gamma(X) = \Gamma(C_{\mathcal{L}}(X))$ for all $X \subseteq \mathcal{L}$.

For a given semantic belief state operator Γ the following belief set operator B_Γ can be introduced: $B_\Gamma(X) = \{Th(K) : K \in \Gamma(X)\}$.

The following examples summarize some types of semantic belief state operators associated to belief set operators investigated in the literature.

Example 10.23 (Poole systems (continued)) Let $\Sigma = (D, E)$, $D \cup E \subseteq \mathcal{L}$ be a Poole system and $Cons_\Sigma(X) = \{\delta : \delta \subseteq D$ and δ is a basis for $X\}$. Let $B_\Sigma(X) = \{Cn(X \cup \delta) : \delta \in Cons_\Sigma(X)\}$. A semantic belief state operator Γ_Σ providing a semantics for B_Σ can be introduced by $\Gamma_\Sigma(X) = \{Mod(T) : T \in B_\Sigma(X)\}$. Obviously, Γ_Σ is Cn-invariant.

Example 10.24 (Generalized belief revision) Let $A \subseteq \mathcal{L}$ be an arbitrary fixed consistent deductively closed set and $X \subseteq \mathcal{L}$ an arbitrary set. Define $Cons(A, X) = \{Y : Y \subseteq A, Y \cup X$ is consistent and Y is maximal with this property$\}$. Let $B(X) = \{Cn(Y \cup X) : Y \in Cons(A, X)\}$. If $A \cup X$ is consistent then $B(X) = \{Cn(A \cup X)\}$. If $A \cup X$ is inconsistent then $B(X)$ contains all complete extensions of X. This can be shown using a generalization of results in [Gro88]. To get belief set operators derived from A, subsets from $Cons(A, X)$ have to be selected. Let $S : \mathcal{P}(\mathcal{L}) \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{L}))$ satisfying $S(X) \subseteq Cons(A, X)$ be such that $S(X) \neq \emptyset$ if $Cons(A, X) \neq \emptyset$. Then the following belief set operator B_S can be introduced: $B_S(X) = \{Cn(Y \cup X) : Y \in S(X)\}$. Again, we may introduce a semantic belief state operator Γ_S for B_S by defining $\Gamma_S(X) = \{Mod(T) : T \in B_S(X)\}$.

Example 10.25 (Stable generated models of logic programs) Stable generated models were introduced in Section 5.3. The following system $(L_{seq}, M, \models, \Gamma)$ is a semantic belief state frame: L_{seq} is the set of sequents, M the set of all partial models, $\models$ the partial satisfiability relation and $\Gamma(P) = \{\{m\} \mid m$ is a stable generated model of $P\}$.

10.4 Representation

The methods described in Section 10.1 can be generalized to the case of belief set operators and semantic belief state frames. In particular, there is a canonical method to introduce a semantics for a given belief set frame.

Proposition 10.26 Let $\mathcal{F} = (\mathcal{L}, C_{\mathcal{L}}, B)$ be a belief set frame satisfying belief left absorption. Then there exists a semantic belief state frame $\mathcal{SF} = (\mathcal{L}, M, \models, \Gamma)$ such that $(\mathcal{L}, C_{\mathcal{L}})$ is complete with respect to $(\mathcal{L}, M, \models)$ and $B = B_\Gamma$. If $\mathcal{F}$ is a deductive belief set frame then $\mathcal{SF}$ can be taken to be $(\mathcal{L}, C_{\mathcal{L}})$-invariant.

Proof: Let $\mathcal{F} = (\mathcal{L}, C_{\mathcal{L}}, B)$ be a belief set frame satisfying belief left absorption. We construct a semantic belief state frame $\mathcal{SF} = (\mathcal{L}, M, \models, \Gamma)$ such that $C_{\mathcal{L}} = C^{\models}$ and $B = B_\Gamma$. Let $(\mathcal{L}, M, \models)$ be the LT-semantics for $(\mathcal{L}, C_{\mathcal{L}})$, and define $\Gamma(X) = \{Mod^{\models}(T) \mid T \in B(X)\}$. Then $B_\Gamma = B$. $B_\Gamma(X) = \{Th(Mod^{\models}(T) \mid T \in B(X)\}$, and since $C_{\mathcal{L}}(T) = T$ for $T \in B(X)$ it follows $Th(Mod^{\models}(T)) = T$, hence $B_\Gamma(X) = B(X)$.

Now assume that $\mathcal{F}$ is a deductive belief set frame. Then $C_{\mathcal{L}}(X) = C_{\mathcal{L}}(Y)$ implies $B(X) = B(Y)$. We show that the semantic belief state operator defined above is invariant. Since $C_{\mathcal{L}}(X) = C_{\mathcal{L}}(C_{\mathcal{L}}(X))$, and by congruence $B(X) = B(C_{\mathcal{L}}(X))$, $\Gamma(X) = \{Mod^{\models}(T) \mid T \in B(X)\} = \Gamma(C_{\mathcal{L}}(X)) = \{Mod^{\models}(T) \mid T \in B(C_{\mathcal{L}}(X))\}$. □

The question arises whether a belief set operator B can be extended to a deductive belief set frame $(\mathcal{L}, C_{\mathcal{L}}, B)$. Of course, there is the following trivial solution: $C_{\mathcal{L}}(X) = X$, which cannot be considered as adequate. We will assume that the desired logic for B should be as close as possible to K_B; i.e. $C_{\mathcal{L}}$ should be maximal below K_B with respect to the following partial ordering between inference operations $C_1, C_2 : C_1 \leq C_2 \Leftrightarrow (\forall X \subseteq \mathcal{L})(C_1(X) \subseteq C_2(X))$. In this fashion, as many conclusions as possible are the result of classical (monotonic) reasoning.

Proposition 10.27 Let B be a belief set operator on $\mathcal{L}$ satisfying strong belief cumulativity. Then there exists a deductive system $(\mathcal{L}, C_{\mathcal{L}})$ such that following conditions are satisfied:

1. $(\mathcal{L}, C_{\mathcal{L}}, B)$ is a deductive belief set frame.
2. If $(\mathcal{L}, C_1, B)$ is a deductive belief set frame then $C_1 \leq C_{\mathcal{L}}$, i.e. $C_{\mathcal{L}}$ is the greatest deductive system for $(\mathcal{L}, B)$.

Proof: Since B is strongly cumulative the inference system $(\mathcal{L}, K_B)$ is cumulative. By the main result in [Die94] there exists a largest deductive operation $C_{\mathcal{L}} \leq K_B$ such that $(\mathcal{L}, C_{\mathcal{L}}, K_B)$ is a deductive inference frame. Since $\mathcal{BF} = (\mathcal{L}, C_{\mathcal{L}}, B)$ is a strong cumulative belief set frame it follows by Proposition 10.19 that $\mathcal{BF}$ is a deductive belief set frame. $\mathcal{BF}$ satisfies the desired properties. $\square$

The semantical approach presented here can be summarized as follows. We start with a belief set operator B on a language $\mathcal{L}$; in the next step we construct a belief set frame $(\mathcal{L}, C_{\mathcal{L}}, B)$ such that the compact logic $(\mathcal{L}, C_{\mathcal{L}})$ satisfies additional properties, e.g. maximality. Then for $(\mathcal{L}, C_{\mathcal{L}}, B)$ we may introduce the standard semantics indicated in Proposition 10.26 (see Figure 10.1).

Belief set operator
$(\mathcal{L}, B)$
$\Downarrow$
Deductive Belief Set Frame
$(\mathcal{L}, C_{\mathcal{L}}, B)$
$\Downarrow$
Semantic Belief State Frame
$(\mathcal{L}, \mathsf{Mod}, \models, \Gamma)$

Figure 10.1: Standard semantics of belief set operators.

Finally, we return to the connections between deductive inference frames and deductive belief set frames. Obviously, as mentioned before, deductive inference frames $(\mathcal{L}, C_{\mathcal{L}}, C)$ can be considered as a special case of belief set frames by taking $B_C(X) = \{C(X)\}$. On the other hand, for every deductive belief set frame $(\mathcal{L}, C_{\mathcal{L}}, B)$ there exists exactly one deductive inference frame defined by the kernel K_B. The converse is not true: for a given deductive inference frame there can be many deductive belief set frames with the same kernel. Belief set frames can be understood as specializations of deductive inference frames and a deductive inference frame can be interpreted as an abstract representation of a family of deductive belief set frames. We will make this precise.

Definition 10.28 Let $\mathcal{F} = (\mathcal{L}, C_{\mathcal{L}}, C)$ be a deductive inference frame.

1. Define $\Omega(\mathcal{F}) = \{B : (\mathcal{L}, C_{\mathcal{L}}, B)$ is a consistency preserving deductive belief set frame such that $C = K_B\}$.

2. The binary relations $\preceq$ and $\equiv$ between belief set operators in $\Omega(\mathcal{F})$ are defined as follows:

$$\begin{array}{lll} B_1 \preceq B_2 & \text{iff} & (\forall X)(B_1(X) \preceq B_2(X)), \quad \text{and} \\ B_1 \equiv B_2 & \text{iff} & B_1 \preceq B_2 \text{ and } B_2 \preceq B_1. \end{array}$$

3. Let $\mathbf{BF}(\mathcal{F}) = (\Omega(\mathcal{F}), \preceq)$ and $Max(X) = \{S : S$ is a maximal consistent extension of $X\}$.

4. A belief set operator $B \in \Omega(\mathcal{F})$ is said to be a *maximization operator* iff $(\forall X \subseteq L)(B(X) \subseteq Max(C(X)))$.

We collect a number of results about the structure of $\mathbf{BF}(\mathcal{F})$.

Proposition 10.29 Let $\mathcal{F} = (\mathcal{L}, C_{\mathcal{L}}, C)$ be a deductive inference frame. Then $\mathbf{BF}(\mathcal{F}) = (\Omega(\mathcal{F}), \preceq)$ is a partial ordering.

Proof: Obviously, the relation $\preceq$ satisfies reflexivity and transitivity. We show anti-symmetry. Assume $B_1 \preceq B_2$ and $B_2 \preceq B_1$ for $B_1, B_2 \in \Omega(\mathcal{F})$. Let $U \in B_1(X)$, by assumption there is a $V \in B_2(X)$ such that $V \subseteq U$; since $B_1(X) \preceq B_2(X)$ there is a set $W \in B_1(X)$ satisfying $W \subseteq V$. Non-inclusiveness of $B_1(X)$ implies $U = V$, hence $U \in B_2(X)$. Analogously one shows $B_2(X) \subseteq B_1(X)$. $\square$

Proposition 10.30 Let $\mathcal{F} = (\mathcal{L}_0, Cn, C)$ be a deductive inference frame over classical logic $(\mathcal{L}_0, Cn)$. The system $\mathbf{BF}(\mathcal{F})$ has a least element and a least maximization operator. A belief set operator $B \in \mathbf{BF}(\mathcal{F})$ is a maximal element with respect to $\preceq$ if and only if B is a maximization operator such that for every $X \subseteq L$ and $T \in B(X)$ the following condition $(*)$ $C(X) \neq \bigcap(B(X) - \{T\})$ is satisfied.

Proof: Let $\mathcal{F} = (\mathcal{L}_0, Cn, C)$; the least element B_{min} of $\Omega(\mathcal{F})$ is defined by $B_{min}(X) = \{C(X)\}$, and the least maximization operator is determined by $B_{max}(X) = Max(C(X))$. Now, let B be a maximal element. We first show that B is a maximization operator. Assume this is not the case. Then there is a belief set $T \in B(X)$ (for a certain set $X \subseteq \mathcal{L}_0$), such that T is not maximal. We define a new operator B_1 as follows: $B_1(Y) = B(Y)$ for all $Y \neq X$, and $B_1(X) = (B(X) - \{T\}) \cup Max(T)$. It is easy to show that $B \preceq B_1$, but not $B_1 \preceq B$. Now we will show $(*)$. Suppose there exist $X \subseteq \mathcal{L}_0$ and $T \in B(X)$ such that $C(X) = \bigcap(B(X) - \{T\})$. Then define B_1 by setting $B_1(Y) = B(Y)$ for all $Y \neq X$, and $B_1(X) = B(X) - \{T\}$. Then $B \preceq B_1 \in \mathbf{BF}(\mathcal{F})$, contradicting maximality of B.

Conversely, assume that B is a maximization operator satisfying the condition $(*)$. Suppose B is not maximal. Then there is an operator $B_1 \in \mathbf{BF}(\mathcal{F})$, such that $B(X) \preceq B_1(X)$, but $B(X) \neq B_1(X)$. Since every $T \in B_1(X)$ is an extension of a belief set of $B(X)$ and every belief set in $B(X)$ is a maximal extension of $C(X)$ it holds that $B_1(X) \subseteq B(X)$. Hence, by condition $(*)$ it follows $\bigcap B_1(X) \neq C(X)$. This gives a contradiction. □

A belief set operator $B \in \Omega(\mathcal{F})$ satisfies *C-congruence* iff $(\forall XY \subseteq \mathcal{L})(C(X) = C(Y) \Rightarrow B(X) = B(Y))$. The following observation is straightforward.

Proposition 10.31 Let $\mathcal{F} = (\mathcal{L}_0, Cn, C)$ be a cumulative deductive inference frame. Then every C-congruent belief set operator in $\Omega(\mathcal{F})$ satisfies belief cumulativity, i.e. weak belief monotony and belief cut. Furthermore, the least maximization operator satisfies C-congruence.

Remark 10.32 Concerning the structure of $\mathbf{BF}(\mathcal{F})$ there is the following question. Let P be a property on belief set frames, and $\mathcal{F}$ is a cumulative deductive inference frame. Does there exist an element in $\mathbf{BF}(\mathcal{F})$ which is maximal with respect to the property P? Examples of such properties are distributivity or strong belief cumulativity.

10.5 Selection operators

In the previous sections we concentrated on the multiple belief set view. The kernel of a belief set operator represents the most certain inferences the agent can make. But there is also another way in which the agent can handle the multiple views, and that is by selecting one (or a subset) of the possible views and focusing on this view. In the area of design, given some requirements a designing agent may have multiple (partial) descriptions of objects that do not contradict the requirements. It may have one of these descriptions (views) in focus, which it will try to complete. Here the selection indicates which view is in focus. On the other hand, for many nonmonotonic formalisms in which a theory can have multiple extensions (or expansions), a prioritized or stratified version exists, in which control knowledge (such as a preference ordering on the nonmonotonic rules) is used to designate one of the extensions as the most preferred one ([Bre94a], [ABW88], [Kon88a], [TT92]). This focusing mechanism can be studied abstractly through *selective inference operations* for a given belief set operator which choose one of the sets of beliefs.

Definition 10.33 (Selective inference operation) Let B be a belief set operator. A selective inference operation for B is an inference operation C such that $\forall X \subseteq \mathcal{L} : C(X) \in B(X)$.

We consider a typical example of a selective inference operation for the belief set operator based on default logic.

Example 10.34 (Prioritized default logic, [Bre94a]) Let D be a countable set of normal defaults, denoted as a/c, and let X be a set of formulae. Earlier we defined the belief set operator $B_D(X)$ collecting all Reiter-extensions of X with respect to D. If X is consistent then, since D contains only normal defaults, the set $B_D(X)$ is non-empty. For every strict total ordering $\ll$ of D we define a selective inference operation $C_{\ll}$ for B_D as follows. A default $\delta = a/c$ is said to be *active* in a set Z of formulae if $a \in Z$, $c \notin Z$, and $\neg c \notin Z$. Let a set X be given and define a sequence $\{E_i : i < \omega\}$ as follows:

$$\begin{array}{lcl} E_0 & = & Cn(X), \text{ and for } i \geq 0 : \\ E_{i+1} & = & \begin{cases} E_i & \text{if no default is active in } E_i; \\ Cn(E_i \cup \{c\}) & \text{otherwise, where c is the consequent} \\ & \text{of the } \ll\text{-least default that is active in } E_i. \end{cases} \end{array}$$

We define $C_{\ll}(X) = \bigcup_{i<\omega} E_i$. It can be shown that $C_{\ll}(X) \in B_D(X)$ ([Bre94a]). The extension $\bigcup_{i=0}^{\infty} E_i$ is called the *prioritized extension of* (D, X) *generated by* $\ll$.

One may argue that the concept of a selective inference operations is already covered by the notion of a usual inference operation. Obviously, this is not the case because a selective inference operation is always connected with a belief set operator as a separate notion. As an example imagine an agent A which acts under incomplete information in a dynamic environment. It is important for A to have an appropriate basic space of different belief sets and an additional mechanism to choose and generate one of them to adapt his behavior to a particular situation. In principle, this idea can also be realized by a suitable family of usual inference operations and a choice mechanism. To structure the connections between belief set operators and selective inference operations, we give the following definition:

Definition 10.35

1. Let a belief set operator B be given. The family of selective inference opera-

tions for B, denoted by $\mathcal{C}_B$, is defined by

$$\mathcal{C}_B = \{C \mid C \text{ is a selective inference operation for } B\}.$$

2. Let $\mathcal{C}$ be a family of inference operations. Define the belief set operator $B_{\mathcal{C}}$ by

$$B_{\mathcal{C}}(X) = \{C(X) \mid C \in \mathcal{C}\}.$$

A selective inference operation for a belief set operator will in general be more informative than the associated kernel: $K_B(X) \subseteq C(X)$. But even if the belief set operator is well-behaved, a selective inference operation can be badly behaved. The question arises whether a well-behaved selective inference operation always exists. That is, given a belief set operator B, the question is whether there exists a $C \in \mathcal{C}_B$ with certain structural properties. This is a hard question. Sufficient conditions can be found, for instance for monotony: $(\forall Y)(\exists T \in B(Y))(\forall X \subseteq Y)(\forall S \in B(X)) : S \subseteq T$. But this condition implies (in the presence of non-inclusiveness) that $B(X)$ is a singleton for all X. Necessary conditions are easier to find, but quite trivial. For a belief set operator B and a selective inference operation C for B we have the following.

If C satisfies cut then
$(\forall X)(\exists S \in B(X))(\forall Y)(X \subseteq Y \subseteq S \Rightarrow ((\exists T \in B(Y))(T \subseteq S))$ (*1).
If C satisfies cautious monotony then
$(\forall X)(\exists S \in B(X))(\forall Y)(X \subseteq Y \subseteq S \Rightarrow ((\exists T \in B(Y))(S \subseteq T))$ (*2).
If C satisfies cumulativity then
$(\forall X)(\exists S \in B(X))(\forall Y)(X \subseteq Y \subseteq S \Rightarrow ((\exists T \in B(Y))(S = T))$ (*3).
If C satisfies monotony then
$(\forall XY)(X \subseteq Y \Rightarrow ((\exists S \in B(X))(\exists T \in B(Y))(S \subseteq T))$ (*4).

The preceding paragraph pertains to the situation when a belief set operator B is given, and we want to study $\mathcal{C}_B$. Questions about the second item in Definition 10.35 are easier to answer. We will say a family $\mathcal{C}$ of inference operations satisfies one of the properties of cut, cautious monotony, cumulativity and monotony if all of the inference operations in $\mathcal{C}$ satisfy this property. Then we have:

Proposition 10.36 Let $\mathcal{C}$ be a family of inference operations.

1. If $\mathcal{C}$ satisfies monotony then $B_{\mathcal{C}}$ satisfies belief monotony.
2. if $\mathcal{C}$ satisfies cautious monotony then $B_{\mathcal{C}}$ satisfies weak belief monotony.
3. if $\mathcal{C}$ satisfies cut, then $B_{\mathcal{C}}$ satisfies both belief transitivity and (strong) belief cut.
4. if $\mathcal{C}$ satisfies cumulativity, then $B_{\mathcal{C}}$ satisfies strong belief cumulativity.

Proof:

1. Suppose $\mathcal{C}$ satisfies monotony, and suppose $X \preceq Y$. Take any $C(Y) \in B_{\mathcal{C}}(Y)$, then $C(X) \subseteq C(Y)$ and $C(X) \in B_{\mathcal{C}}(X)$. We have $B_{\mathcal{C}}(X) \preceq B_{\mathcal{C}}(Y)$.

2. Suppose $X \preceq Y \preceq B_{\mathcal{C}}(X)$. Take a $C(Y) \in B_{\mathcal{C}}(Y)$, then $X \subseteq Y \subseteq C(X)$ (since $Y \preceq B_{\mathcal{C}}(X)$), so $C(X) \subseteq C(Y)$. It again follows that $B_{\mathcal{C}}(X) \preceq B_{\mathcal{C}}(Y)$.

3. Suppose $\mathcal{C}$ satisfies cut. We only have to prove that $B_{\mathcal{C}}$ satisfies strong belief cut. So suppose $C(X) \in B_{\mathcal{C}}(X)$ and $X \subseteq Y \subseteq C(X)$. Then $C(Y) \subseteq C(X)$ and $C(Y) \in B_{\mathcal{C}}(Y)$.

4. If $\mathcal{C}$ satisfies cumulativity, it satisfies cautious monotony and cut, so by 2. and 3. $B_{\mathcal{C}}$ satisfies weak belief monotony, belief cut and belief transitivity, hence it satisfies strong belief cumulativity.

□

One way of defining selective inference operations for a given belief set operator is through *selection operators*. Given a set of views, such a selection operator selects one (or some) of them. These selection operators were already defined in general in Section 2.2.

Definition 10.37 A selection operator is a function $s : \mathcal{P}(\mathcal{P}(\mathcal{L})) \to \mathcal{P}(\mathcal{P}(\mathcal{L}))$ such that for all $\mathcal{A} \subseteq \mathcal{P}(\mathcal{L}) : s(\mathcal{A}) \subseteq \mathcal{A}$, and $s(\mathcal{A}) \neq \emptyset$ if $\mathcal{A} \neq \emptyset$. The operator s is single-valued iff for all non-empty $\mathcal{A} \subseteq \mathcal{P}(\mathcal{L})$: $\#(s(\mathcal{A})) = 1$. A single-valued selection function s can be understood as a choice function $s : \mathcal{P}(\mathcal{P}(\mathcal{L})) \to \mathcal{P}(\mathcal{L})$ satisfying $s(\mathcal{A}) \in \mathcal{A}$ for all non-empty $\mathcal{A}$.

Using selection operators we can generate inference operations:

Definition 10.38 Let a belief set operator B and a selection operator s be given. We define the inference operation C_s^B by: $C_s^B(X) = \bigcap s(B(X))$.

We will give some examples of belief set operators with selection operators.

Example 10.39 (Autoepistemic logic and parsimonious expansions) It is well-known that in autoepistemic logic it may happen that the objective (i.e., non-modal) part of a stable expansion is contained in the objective part of another stable expansion. The easiest example is the theory $\{Lp \to p\}$, which has two stable expansions: the (unique) stable set with objective part $Cn(\emptyset)$, and the stable set with objective part $Cn(\{p\})$. Given a modal language L_m we can define the belief set operator B_{ael} which assigns to each set I of modal formulae the set of stable expansions of I. But an agent may want to keep only those expansions with a minimal objective part (these are called *parsimonious expansions* in [EG92]). We could define the selection operator s_p by $s_p(\mathcal{A}) = \{X \in \mathcal{A} \mid$ there is no $Y \in \mathcal{A}$ such that the objective part of Y is included in the objective part of $X\}$. Then $s_p(B_{ael}(I))$ is the collection of parsimonious expansions of I, and $C_{s_p}^{B_{ael}}$ gives the (skeptical) conclusions based on these expansions.

Example 10.40 (Prioritized default logic, continued) In Example 10.34, a single extension was selected from the set of all extensions on the basis of a total ordering $\ll$ on the set of defaults D. Often, the priority information will be partial, and we can select the extensions which comply with this partial information (see [Bre94a]). Given a partial ordering $<$ on D, we can define a selection operator that selects those extensions of (D, X) which are generated by a total ordering $\ll$ that extends $<$ (meaning that $d_1 < d_2$ implies $d_1 \ll d_2$).

Another example is the selection operator s_{user} that selects a maximal indicative subset from the output of the multi-interpretation operator MI_{maxind} of Section 8.2.

Single-valued selection operators generate *selective* inference operations. A first observation about when a selective inference operation can be generated by a single-valued selection operator:

Proposition 10.41 Let a selective inference operation C for a belief set operator B be given. Then $C = C_s^B$ for some single-valued selection operator s iff $(\forall X \forall Y)(B(X) = B(Y) \Rightarrow C(X) = C(Y))$.

Proof: Define s as follows: for $\mathcal{A} \subseteq \mathcal{P}(\mathcal{L})$, if $\mathcal{A} = B(X)$ for some $X \subseteq \mathcal{L}$, then $s(\mathcal{A}) = \{C(X)\}$, and if not, then s selects any set from $\mathcal{A}$ (and $s(\emptyset) = \emptyset$). The requirement ensures that s is well-defined, and it is easy to see that s is a single-valued selection operator. For any $X \subseteq \mathcal{L}$ we have $C_s^B(X) = \bigcap s(B(X)) = \bigcap\{C(X)\} = C(X)$. The other direction is trivial. □

We can study properties of selection operators and the relation with properties of belief state operators and selective inference operations. Although a full treatment is beyond the scope of this chapter, we will give an example.

Definition 10.42 A selection operator s satisfies *selection monotony* if for all belief set families $\mathcal{A}$, $\mathcal{B}$ we have $\mathcal{A} \preceq \mathcal{B} \Rightarrow s(\mathcal{A}) \preceq s(\mathcal{B})$.

Then we have the following:

Proposition 10.43

1. Let a belief set operator B and a selection operator s be given. If B satisfies belief monotony and s satisfies selection monotony then C_s^B satisfies monotony.
2. Let a single-valued selection operator s be given. If for any belief set operator B

which satisfies belief monotony, C_s^B satisfies monotony, then s satisfies selection monotony.

Proof:

1. If $X \subseteq Y$ then $B(X) \preceq B(Y)$ (belief monotony) so $s(B(X)) \preceq s(B(Y))$ (selection monotony) so $C_s^B(X) \subseteq C_s(Y)$.

2. Suppose we have two belief set families $\mathcal{A} \preceq \mathcal{B}$. Define a belief set operator B by $B(\emptyset) = \mathcal{A}$ and $B(X) = \mathcal{B}$ for $X \neq \emptyset$. It is easy to see that B satisfies belief monotony. Then as $\emptyset \subseteq \mathcal{L}$, we must have $C_s^B(\emptyset) \subseteq C_s^B(\mathcal{L})$, and as s is single-valued this means that $s(B(\emptyset)) \preceq s(B(\mathcal{L}))$ or $s(\mathcal{A}) \preceq s(\mathcal{B})$.

□

The problem with selection operators is that they are blind to the initial facts: if $B(X) = B(Y)$, then we may sometimes want to make a different selection from $B(X)$ than from $B(Y)$. One option would be to define selection operators s_X with an index for the initial facts. An inference operation $C_s^B(X)$ could then be defined by $C_s^B(X) = \bigcap s_X(B(X))$. This would yield results similar to the construction of $B_{\mathcal{C}}$ defined earlier.

Example 10.44 (Contraction functions) In [AGM85], eight rationality postulates are given for contraction functions. A contraction function $\dot{-}$ is a function that given a belief set K and a formula φ yields a new belief set $K \dot{-} \varphi$ which is meant to be the result of 'removing' φ from K. We list the eight AGM postulates below.

1. For any sentence φ and belief set K closed under propositional provability, the set $K \dot{-} \varphi$ is also closed under propositional provability.

2. $K \dot{-} \varphi \subseteq K$.

3. If $\varphi \notin K$, then $K = K \dot{-} \varphi$.

4. If $\not\models \varphi$ then $\varphi \notin K \dot{-} \varphi$.

5. $K \subseteq (K \dot{-} \varphi) + \varphi$.

6. If $\models \varphi \leftrightarrow \psi$, then $K \dot{-} \varphi = K \dot{-} \psi$.

7. $K \dot{-} \varphi \cap K \dot{-} \psi \subseteq K \dot{-} (\varphi \wedge \psi)$.

8. If $\varphi \notin K \dot{-} (\varphi \wedge \psi)$, then $K \dot{-} (\varphi \wedge \psi) \subseteq K \dot{-} \psi$.

Call a selection operator s_X *invariant* if $s_X = s_{Cn(X)}$ for all X. Then a result from [AGM85] can be given in our terms:
A contraction function $\dot{-}$ satisfies the first 6 postulates iff $X \dot{-} \varphi = C_s^{\Gamma - \varphi}(X)$ for

some invariant s, where $\Gamma_{-\varphi}$ is the belief set operator defined in Example 2.14. Furthermore, if we put extra conditions on the selection operator — intuitively, that it picks maximal elements from $\Gamma_{-\varphi}$ given some transitive and reflexive order — then this result can be strengthened in the sense that all rationality postulates hold.

Remark: The considerations in the Sections 10.4 and 10.5 reflect certain aspects of knowledge dynamics [Pop77]. Let X_0 be a deductively closed set representing the knowledge at a certain time point. X_0 can be extended by a combined application of a belief set operator B_0 whose kernel is X_0 and a selection operator s_0. The new knowledge stage X_1 is defined by $X_1 = C_{s_0}^{B_0}(X_0) = \bigcap s_0(B_0(X_0))$. The forming of belief sets for a knowledge base can be understood as theory formation or hypothesis building; after new observations are performed those belief sets are left out which contradict the observations.

10.6 Conclusions and related work

In this chapter, we have analyzed belief set operators, which can be seen as syntactical multiple belief state operators. They are a generalization of the well-known inference operations. Some properties of belief set operators (generalized from the inference operation properties) were defined and investigated. A general semantic counterpart for a belief set operator was treated, and it was shown that for a class of belief set operators there is a generic way of defining a maximal underlying semantics. The selection operators already introduced in Chapter 2 were applied to belief set operators.

The idea of investigating belief set operators was already present in [Mak94] and [Voo93]. The present chapter takes some preliminary steps toward a full study of belief set operators, in the spirit of what has been done for inference operations.

Acknowledgments

The material of this chapter appeared earlier in [EHT96].

Chapter 11

Conclusions and Perspectives

Throughout this thesis, we have given detailed conclusions and we have discussed related work and occasionally further research issues at the end of each chapter, or sometimes section. Therefore, we will not do that here. Rather, we will give a broad overview of what we have done in this thesis, and sketch a general perspective for possible future research in this area.

The most general goal of this work has been to convince the reader that it is possible, interesting, worthwhile, and sometimes even necessary to analyze complex reasoning on a level of abstraction in-between the level of inference operations and the level of system specification. In particular, the nondeterministic and dynamic aspects of reasoning can and should be modeled and studied. To this end, we introduced a hierarchy of levels of abstraction for specifying and analyzing reasoning in Chapter 1. The first and last level are well-known, but the two intermediate levels are new (although hinted at in the literature). The exact mathematical formalization of reasoning at the first three levels, in terms of final belief state operators (level 1), multiple belief state operators (level 2) and reasoning frame operators (level 3), was given in Chapter 2.

The larger part of the rest of this thesis concentrated on level 3, with the exception of Section 8.2 and Chapter 10. The latter chapter gave a preliminary analysis of syntactic multiple belief state operators in terms of possible properties, underlying semantics and links with level 1. A possible use of (a slight variant of) these operators for formalizing a practical reasoning task was given in Section 8.2. No specification languages for this level were proposed (although a first proposal is in [ET96c]).

We have advocated and hopefully made the reader interested in the dynamic viewpoint on commonsense reasoning (this is done in [Ben96a] in the field of Logic). The dynamic view on default logic and logic programming was introduced in Chapter 3 by showing that they can be viewed as specification languages for reasoning frame operators. We proposed epistemic variants of temporal logics (or temporal variants of epistemic logics) as natural candidates for a specification language for informational behavior in Chapter 4.

Then in Chapter 5 we specified a number of (known) forms of reasoning in our temporal logics. This showed at least two things. First, that a dynamic view on these forms of reasoning is possibly in a meaningful way (for default reasoning and normal logic programming, this was already shown in Chapter 3). Second, it shows that our temporal logics are at least powerful enough to, but also natural for, specifying reasoning.

The expressiveness of (one of our variants of) temporal logic and a variant of default logic in a more traditional mathematical sense was investigated in Chapter 7. Both languages are expressively complete for a natural class of reasoning frame operators (and multiple belief state operators) when allowing infinite constructs in the language.

In Chapter 6 it was shown that a useful fragment of one of our temporal specification languages is executable. This means that there is an algorithm that automatically performs the reasoning specified by a temporal theory. In other words, it finds the temporal models of a specification in temporal logic. We gave a generic task model in DESIRE capable of performing this algorithm. The generic task model is easy to extend when augmenting the temporal specification language.

As an example of the use of temporal specification, in Section 8.1 it was shown that it is possible to specify compositional multi-agent systems in a temporal partial logic. This logic can be used to formalize compositional verification and validation proofs for such systems.

Finally, some logical themes inspired by the logic MTEL were the subject of Chapter 9. Axiom systems for TEL and TELC were given in Section 9.1, and decidability and complexity of these logics were investigated. MTEL was used as an example in Section 9.2 where it was shown that in preferential logics formulae may exist that allow us to keep some of the advantages of monotonicity. The inference relation of that same logic is also an example of a non-cumulative relation; non-cumulative inference relations were the subject of Section 9.3.

Let us now look at perspectives for further research. One important direction has been described extensively at the end of Chapter 4: investigating different temporal logics of information (different underlying logics, different time structures, different languages, etc.).

Secondly, the analysis of multiple belief state operators in this thesis has not been extensive. So there is a lot of work that could be done in that area. A further analysis of properties of MBSOs, their interrelationships, the connection with properties of FBSOs would complement the work done in this area for inference relations. Another question is whether and how preferential semantics can be used to give semantics to MBSOs. Furthermore, it would be interesting to consider known formalisms, like default logic, and identify which properties their associated MBSOs have, and which they do not have. These (refined) properties of MBSOs may help to explain the properties the associated FBSO has or does not have. Lastly, we have investigated specification languages for level 3. The same could be done for level 2.

A large area of further research lies in incorporating aspects of reasoning which are inherently dynamic. At the beginning of this thesis (Section 1.1) we gave an example of the kind of reasoning processes we were interested in. It involved an agent that wanted to buy a ticket for a movie, and had to make assumptions, revise its knowledge, reason about observations and perform observations. Although we have laid the foundations that allow us to model this kind of reasoning, and we have shown that many kinds of defeasible reasoning can be specified, an integration of these aspects into one formalism (possibly a family of formalisms) would be very useful. We would like an integrated account of the reasoning process of an agent that performs nonmonotonic reasoning, belief revision, that can update its beliefs as a result of observations and communication, that may reason about actions which it then can perform, etcetera. Also, it would be useful if the agent can reason about time itself. This is needed for resource-bounded reasoning, where time is a resource. In certain situations, the reasoning process of an agent may be adjusted because it realizes that there is not much time in which to make a decision (this is also a phenomenon that active logics are meant to be able to model; see [NKP94]).

In this thesis, there was no notion of a real external world (including other agents) evolving while the agent was reasoning, and this is certainly necessary for practical reasoning. This also touches the point of the number of agents, another area of further research. The extension of many concepts and results in this thesis to the more than one agent case will doubtlessly (much) increase the complexity. In the multi-agent case, however, the dynamics of reasoning, including communication, are even more of influence to the final conclusions reached, if this is even a well-defined or interesting notion. An analysis of multi-agent reasoning on level 1 or 2 is not unthinkable (and would be an interesting subject in certain cases), but the dynamic stance is even more important here than in the single-agent case reported in this thesis.

The usefulness of a temporal logic of information for verification and validation of compositional multi-agent systems (see Section 8.1) would be improved if it had explicit provisions for the multi-agent case. In Section 8.1 we just used different signatures for the different agents (and different components within agents). A more structured approach would be to add different modal operators for each component. This has been worked out in [EJT98]. The temporal semantics in that paper, however, are still 'flat', and modal operators cannot be nested. It would be even better if the compositional structure of the system were reflected in a compositional semantics. This means that the temporal models of a component (including the top component) should be composed of the temporal models of its subcomponents. In order for temporal logics of multi-agent systems to be useful in practice, attention should be paid to proof systems and executability (for prototyping). The verification and validation process should be supported by tools that aid the user in building and managing compositional proofs.

This thesis is by no means a study of all relevant aspects of the nondeterminism and dynamics of practical reasoning. But we hope to have shown the importance of these issues, and the possibility of studying them in an abstract setting. Much more research is needed to obtain a unified view on practical reasoning that integrates many known forms of reasoning.

Bibliography

[ABN95] H. Andréka, J.F.A.K. van Benthem, and I. Németi. Back and forth between modal logic and classical logic. *Journal of the Interest Group in Pure and Applied Logics*, 3(5):685–720, 1995.

[ABW88] K.R. Apt, H.A. Blair, and A. Walker. Towards a theory of declarative knowledge. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 89–142. Morgan Kaufmann, 1988.

[ACGP96] G. Amati, L. Carlucci Aiello, D.M. Gabbay, and F. Pirri. A structural property on modal frames characterizing default logic. *Journal of the IGPL*, 4(1):1–24, 1996.

[ACP97] G. Amati, L. Carlucci Aiello, and F. Pirri. Definability and commonsense reasoning. *Artificial Intelligence*, 93(1–2):169–199, 1997.

[AGM85] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[AL93] M. Abadi and L. Lamport. Composing specifications. *ACM Transactions on Programming Languages and Systems*, 15(1):73–132, 1993.

[Apt90] K.R. Apt. Logic programming. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 493–574. MIT Press, 1990.

[BBST98] F. van Beusekom, F.M.T. Brazier, P. Schipper, and J. Treur. Development of an ecological decision support system. In A.P. del Pobil, J. Mira, and M. Ali, editors, *Tasks and Methods in Applied Artificial Intelligence, Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artifical Intelligence and Expert Systems, IEA/AIE'98, Vol. II*, volume 1416 of *Lecture Notes in Artificial Intelligence*, pages 815–825. Springer-Verlag, 1998.

[BCG+98] F.M.T. Brazier, F. Cornelissen, R. Gustavsson, C.M. Jonker, O. Lindeberg, B. Polak, and J. Treur. Compositional design and verification of a multi-agent system for one-to-many negotiation. In *Proceedings of the*

Third International Conference on Multi-Agent Systems, ICMAS'98, pages 49–56. IEEE Computer Society Press, 1998.

[BDJT95] F.M.T. Brazier, B.M. Dunin-Keplicz, N.R. Jennings, and J. Treur. Formal specification of multi-agent systems: a real world case. In V. Lesser, editor, *Proceedings First International Conference on Multi-Agent Systems, ICMAS'95*, pages 25–32. MIT Press, 1995. Extended version appeared as "DESIRE: modelling multi-agent systems in a compositional formal framework" in M. Huhns and M. Singh, editors, *International Journal of Co-operative Information Systems*, 6(1):67–94, 1997. Special issue on Formal Methods in Co-operative Information Systems: Multi-Agent Systems.

[Ben85] J.F.A.K. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Napoli and Atlantic Heights (N.J.), 1985.

[Ben89] J.F.A.K. van Benthem. Semantic parallels in natural language and computation. In H.D. Ebbinghaus et al., editors, *Logic Colloquium '87*, pages 331–375, 1989.

[Ben91a] J.F.A.K. van Benthem. Logic and the flow of information. In D. Prawitz, B. Skyrms, and D. Westerstahl, editors, *Proceedings 9th International Congress of Logic, Methodology and Philosophy of Science*. North-Holland, Amsterdam, 1991.

[Ben91b] J.F.A.K. van Benthem. *The Logic of Time: A Model-theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*. Kluwer Academic Publishers, Dordrecht, second edition, 1991.

[Ben96a] J.F.A.K. van Benthem. *Exploring Logical Dynamics*. CSLI Publications, Stanford University, 1996.

[Ben96b] J.F.A.K. van Benthem. Temporal logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. IV: Epistemic and Temporal Reasoning*, pages 241–350. Oxford University Press, 1996.

[Bes89] P. Besnard. *An Introduction to Default Logic*. Springer-Verlag, 1989.

[BET98] F.M.T. Brazier, J. Engelfriet, and J. Treur. Analysis of multi-interpretable ecological monitoring information. In A. Hunter and S. Parsons, editors, *Applications of Uncertainty Formalisms*, volume 1455 of *Lecture Notes in Artificial Intelligence*, pages 303–324. Springer-Verlag, 1998.

[BFG+96] H. Barringer, M. Fisher, D.M. Gabbay, R. Owens, and M. Reynolds. *The Imperative Future: Principles of Executable Temporal Logic*. Research Studies Press Ltd. and John Wiley & Sons, 1996.

[BFGH91] H. Barringer, M. Fisher, D.M. Gabbay, and A. Hunter. Meta-reasoning in executable temporal logic. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, KR'91*, 1991.

[BJT98] F.M.T. Brazier, C.M. Jonker, and J. Treur. Principles of compositional multi-agent system development. In J. Cuena, editor, *Proceedings of the IFIP'98 Conference on Information Technology and Knowledge Systems, IT&Knows'98*, pages 347–360. Chapman and Hall, 1998.

[BK82] K.A. Bowen and R. Kowalski. Amalgamating language and meta-language in logic programming. In K. Clark and S. Tarnlund, editors, *Logic Programming*. Academic Press, 1982.

[BK93] A.J. Bonner and M. Kifer. Transaction logic programming. In *Proceedings of the Tenth International Conference on Logic Programming (ICLP)*, pages 257–279. MIT Press, 1993.

[BK95] A.J. Bonner and M. Kifer. Transaction logic programming (or, a logic of procedural and declarative knowledge). Technical Report CSRI-323, Computer Systems Research Institute, University of Toronto, November 1995.

[BL92] H. Bestougeff and G. Ligozat. *Logical tools for temporal knowledge representation*. Ellis Horwood, 1992.

[Bla86] S. Blamey. Partial logic. In D.M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, Vol. III*, pages 1–70. Reidel, Dordrecht, 1986.

[BLRT94] F.M.T. Brazier, P.H.G. van Langen, Zs. Ruttkay, and J. Treur. On formal specification of design tasks. In J.S. Gero and F. Sudweeks, editors, *Artificial Intelligence in Design '94*, pages 535–552. Kluwer Academic Publishers, Dordrecht, 1994.

[BLT96] F.M.T. Brazier, P.H.G. van Langen, and J. Treur. A logical theory of design. In J.S. Gero, editor, *Advances in Formal Design Methods for CAD*, pages 243–266. Chapman and Hall, New York, 1996.

[BM92] P. Besnard and R.E. Mercer. Non-monotonic logics: a valuations-based approach. In B. du Boulay and V. Sgurev, editors, *Artificial Intelligence V: Methodology, Systems, Applications*, pages 77–84. Elsevier Science Publishers, 1992.

[BO97] P.A. Bonatti and N. Olivetti. A sequent calculus for skepical default logic. In D. Galmiche, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, Proceedings TABLEAUX'97*, volume 1227 of *Lecture Notes in Artificial Intelligence*, pages 107–121. Springer-Verlag, 1997.

[Bon96] P.A. Bonatti. Sequent calculi for default and autoepistemic logics. In P. Miglioli, editor, *Theorem Proving with Analytic Tableaux and Related Methods, Proceedings TABLEAUX'96*, volume 1071 of *Lecture Notes in Artificial Intelligence*, pages 127–142. Springer-Verlag, 1996.

[Bou94a] C. Boutilier. Conditional logics of normality: A modal approach. *Artificial Intelligence*, 68:87–154, 1994.

[Bou94b] C. Boutilier. Unifying default reasoning and belief revision in a modal framework. *Artificial Intelligence*, 68:33–85, 1994.

[BPM83] M. Ben-Ari, A. Pnueli, and Z. Manna. The temporal logic of branching time. *Acta Informatica*, 20:207–226, 1983.

[Bra87] M.E. Bratman. *Intentions, Plans, and Practical Reason.* Harvard University Press, Cambridge, USA, 1987.

[Bre91] G. Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense.* Cambridge University Press, 1991.

[Bre94a] G. Brewka. Adding priorities and specificity to default logic. In C. MacNish, D. Pearce, and L.M. Pereira, editors, *Logics in Artificial Intelligence, Proceedings of the 4th European Workshop on Logics in Artificial Intelligence, JELIA'94*, volume 838 of *Lecture Notes in Artificial Intelligence*, pages 247–260. Springer-Verlag, 1994.

[Bre94b] G. Brewka. Reasoning about priorities in default logic. In *Proceedings of the 12th National Conference on Artificial Intelligence, AAAI-94.* MIT Press, 1994.

[BRR89] J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors. *Linear Time, Branching Time and Partial Orders in Logics and Models for Concurrency*, volume 354 of *Lecture Notes in Computer Science.* Springer-Verlag, 1989.

[BS94] P. Besnard and T. Schaub. Possible worlds semantics for default logics. *Fundamenta Informaticae*, 21:39–66, 1994.

[BTWW95] F.M.T. Brazier, J. Treur, N.J.E. Wijngaards, and M. Willems. Formal specification of hierarchically (de)composed tasks. In B.R. Gaines and M.A. Musen, editors, *Proceedings of the 9th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'95*, pages 25/1–25/20. SRDG Publications, Department of Computer Science, University of Calgary, 1995.

[BTWW96] F.M.T. Brazier, J. Treur, N.J.E. Wijngaards, and M. Willems. Temporal semantics of complex reasoning tasks. In B.R. Gaines and M.A. Musen, editors, *Proceedings of the 10th Banff Knowledge Acquisition* *for Knowledge-based Systems workshop, KAW'96*, pages 15/1–15/17. SRDG Publications, Department of Computer Science, University of Calgary, 1996. Extended version to appear in: *Data and Knowledge Engineering*, 1998.

[BW90] J. C. M. Baeten and W. P. Weijland. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 1990.

[Cai78] X. Caicedo. A formal system for the non-theorems of the propositional calculus. *Notre Dame Journal of Formal Logic*, 19:147–151, 1978.

[CB88] W.J. Clancey and C. Bock. Representing control knowledge as abstract tasks and metarules. In Bolc and Coombs, editors, *Expert System Applications*, 1988.

[CES83] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. In *Proceedings of the Tenth Annual ACM Symposium on Principles of Programming Languages*, pages 117–126, 1983. Also appeared in *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[Cha95] E. Chang. *Compositional Verification of Reactive and Real-Time Systems*. PhD thesis, Stanford University, 1995.

[Che80] B.F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, 1980.

[Che97] J. Chen. The generalized logic of only knowing (GOL) that covers the notion of epistemic specifications. *Journal of Logic and Computation*, 7(2):159–174, 1997.

[CJT97] F. Cornelissen, C.M. Jonker, and J. Treur. Compositional verification of knowledge-based systems: a case study for diagnostic reasoning. In E. Plaza and R. Benjamins, editors, *Knowledge Acquisition, Modelling and Management, Proceedings of the 10th EKAW*, volume 1319 of *Lecture Notes in Artificial Intelligence*, pages 65–80. Springer-Verlag, 1997.

[CK90] C. C. Chang and H. J. Keisler. *Model Theory*. North-Holland, Amsterdam, 3rd edition, 1990.

[Cla78] K.L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data-bases*, pages 293–322. Plenum, New York, 1978.

[CMT96] P. Cholewinski, V.W. Marek, and M. Truszczyński. Default reasoning system DeReS. In *Proceedings 5th International Conference on Principles of Knowledge Representation and Reasoning, KR-96*. Morgan Kaufmann, 1996.

[Dav80a] M. Davis. The mathematics of non-monotonic reasoning. *Artificial Intelligence*, 13:73–80, 1980.

[Dav80b] R. Davis. Metarules: reasoning about control. *Artificial Intelligence*, 15:179–222, 1980.

[DGK96] D. Dams, R. Gerth, and P. Kelb. Practical symbolic model checking of the full μ-calculus using compositional abstractions. Technical report, Eindhoven University of Technology, Department of Mathematics and Computer Science, 1996.

[DH94] J. Dietrich and H. Herre. Outline of nonmonotonic model theory. Technical Report NTZ-Report 5/94, Universität Leipzig, Naturw.-Technische Zentrum, Leipzig, 1994.

[Die94] J. Dietrich. Deductive bases of nonmonotonic inference operations. Technical Report NTZ-Report 7/94, Universität Leipzig, Naturw.-Technische Zentrum, Leipzig, 1994.

[Dij76] E.W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[DNR97] F.M. Donini, D. Nardi, and R. Rosati. Ground nonmonotonic modal logics. *Journal of Logic and Computation*, 7(4):523–548, 1997.

[EC82] E.A. Emerson and E.M. Clarke. Using branching time logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2:241–266, 1982.

[EEF+98] P.A.T. van Eck, J. Engelfriet, D. Fensel, F. van Harmelen, Y. Venema, and M. Willems. Specification of dynamics for knowledge-based systems (extended abstract). In B. Freitag, H. Decker, M. Kifer, and A. Voronkov, editors, *Transactions and Change in Logic Databases*, volume 1472 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998. In Press.

[EG92] T. Eiter and G. Gottlob. Reasoning with parsimonious and moderately grounded expansions. *Fundamenta Informaticae*, 17:31–53, 1992.

[EH97] J. Engelfriet and H. Herre. Generated models and extensions of nonmonotonic systems. In J. Małuszyński, editor, *Logic Programming, Proceedings of the 1997 International Symposium, ILPS-97*, pages 85–99. MIT Press, 1997.

[EHT95] J. Engelfriet, H. Herre, and J. Treur. Nonmonotonic belief state frames and reasoning frames (extended abstract). In C. Froidevaux and J. Kohlas, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Proceedings ECSQARU'95*, volume 946 of *Lecture Notes in Artificial Intelligence*, pages 189–196. Springer-Verlag, 1995.

[EHT96] J. Engelfriet, H. Herre, and J. Treur. Nonmonotonic reasoning with multiple belief sets. In D.M. Gabbay and H.J. Ohlbach, editors, *Practical Reasoning, International Conference on Formal and Applied Practical Reasoning, FAPR'96*, volume 1085 of *Lecture Notes in Artificial Intelligence*, pages 331–344. Springer-Verlag, 1996. Revised version to appear in the *Annals of Mathematics and Artificial Intelligence.*

[EJT98] J. Engelfriet, C.M. Jonker, and J. Treur. Compositional verification of multi-agent systems in temporal multi-epistemic logic. In J.P. Müller, M.P. Singh, and A.S. Rao, editors, *Intelligent Agents V*, Lecture Notes in Artificial Intelligence, pages 177–194. Springer-Verlag, 1998.

[Elg88] J.J. Elgot-Drapkin. *Step–logic: Reasoning Situated in Time.* PhD thesis, Faculty of the Graduate School of the University of Maryland, 1988.

[Eme90] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 996–1072. Elsevier Science Publishers, 1990.

[EMTT96] J. Engelfriet, V.W. Marek, J. Treur, and M. Truszczyński. Infinitary default logic for specification of nonmonotonic reasoning. In J.J. Alferes, L.M. Pereira, and E. Orlowska, editors, *Logics in Artificial Intelligence, Proceedings European Workshop on Logics in Artificial Intelligence, JELIA'96*, volume 1126 of *Lecture Notes in Artificial Intelligence*, pages 224–236. Springer-Verlag, 1996.

[Eng] J. Engelfriet. Non-cumulative reasoning: Rules and models. Submitted.

[Eng96a] J. Engelfriet. Minimal temporal epistemic logic. *Notre Dame Journal of Formal Logic*, 37(2):233–259, 1996. Special issue on combining logics, edited by M. de Rijke and P. Blackburn.

[Eng96b] J. Engelfriet. Only persistence makes nonmonotonicity monotonous (extended abstract). In J.J. Alferes, L.M. Pereira, and E. Orlowska, editors, *Logics in Artificial Intelligence, Proceedings European Workshop on Logics in Artificial Intelligence, JELIA'96*, volume 1126 of *Lecture Notes in Artificial Intelligence*, pages 164–175. Springer-Verlag, 1996.

[Eng98] J. Engelfriet. Monotonicity and persistence in preferential logics. *Journal of Artificial Intelligence Research*, 8:1–21, 1998.

[EP90] J.J. Elgot-Drapkin and D. Perlis. Reasoning situated in time I: Basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence*, 2:75–98, 1990.

[ES89] E.A. Emerson and J. Srinivasan. Branching time temporal logic. In Bakker et al. [BRR89], pages 123–172.

[ET93] J. Engelfriet and J. Treur. A temporal model theory for default logic. In M. Clarke, R. Kruse, and S. Moral, editors, *Proceedings 2nd European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, ECSQARU '93*, volume 747 of *Lecture Notes in Computer Science*, pages 91–96. Springer-Verlag, 1993. A fully revised and extended version appeared as "An interpretation of default logic in minimal temporal epistemic logic", *Journal of Logic, Language and Information*, 7:369–388, 1998.

[ET94a] J. Engelfriet and J. Treur. Relating linear and branching time temporal models. Technical Report IR-353, Vrije Universiteit Amsterdam, Faculty of Mathematics and Computer Science, 1994. A revised version is being prepared for submission.

[ET94b] J. Engelfriet and J. Treur. Temporal theories of reasoning. In C. MacNish, D. Pearce, and L.M. Pereira, editors, *Logics in Artificial Intelligence, Proceedings of the 4th European Workshop on Logics in Artificial Inteligence, JELIA'94*, volume 838 of *Lecture Notes in Artificial Intelligence*, pages 279–299. Springer-Verlag, 1994. Also appeared in the *Journal of Applied Non–Classical Logics*, 5(2):239–261, 1995.

[ET96a] J. Engelfriet and J. Treur. Executable temporal logic for nonmonotonic reasoning. *Journal of Symbolic Computation*, 22(5 & 6):615–625, 1996. Special issue on executable temporal logics, edited by M. Fisher, S. Kono, and M.A. Orgun.

[ET96b] J. Engelfriet and J. Treur. Semantics for default logic based on specific branching time models. In W. Wahlster, editor, *Proceedings 12th European Conference on Artificial Intelligence, ECAI'96*, pages 60–64. John Wiley & Sons, 1996.

[ET96c] J. Engelfriet and J. Treur. Specification of nonmonotonic reasoning. In D.M. Gabbay and H.J. Ohlbach, editors, *Practical Reasoning, International Conference on Formal and Applied Practical Reasoning, FAPR'96*, volume 1085 of *Lecture Notes in Artificial Intelligence*, pages 111–125. Springer-Verlag, 1996.

[ET97] J. Engelfriet and J. Treur. A compositional reasoning system for executing nonmonotonic theories of reasoning. In D.M. Gabbay, R. Kruse, A. Nonnengart, and H.J. Ohlbach, editors, *Qualitative and Quantitative Practical Reasoning, Proceedings ECSQARU-FAPR'97*, volume 1244 of *Lecture Notes in Artificial Intelligence*, pages 252–266. Springer-Verlag, 1997.

[ET98] J. Engelfriet and J. Treur. Multi-interpretation operators and approximate classification. Manuscript, 1998.

[Eth87] D.W. Etherington. A semantics for default logic. In *Proceedings 10th International Joint Conference on Artificial Intelligence*, 1987.

[Eth88] D. W. Etherington. *Reasoning with Incomplete Information.* Morgan Kaufmann Publishers, Los Altos, CA, 1988.

[EV98] J. Engelfriet and Y. Venema. A modal logic of information change. In I. Gilboa, editor, *Theoretical Aspects of Rationality and Knowledge, Proceedings of the Seventh Conference (TARK 1998)*, pages 125–131. Morgan Kaufmann, 1998.

[Fer91] A. Ferry. Enriched nonmonotonic rule system. Master's thesis, University of Kentucky, 1991.

[Fer94] A. Ferry. *Topological Characterizations for Logic Programming Semantics.* PhD thesis, University of Michigan, 1994.

[FG92] M. Finger and D.M. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 1:203–233, 1992.

[FG96] M. Finger and D.M. Gabbay. Combining temporal logic systems. *Notre Dame Journal of Formal Logic*, 37(2):204–232, 1996. Special Issue on Combining Logics. Edited by M. de Rijke and P. Blackburn.

[FHMV95] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning About Knowledge.* MIT Press, 1995.

[Fis94] M. Fisher. A survey of concurrent MetateM — the language and its applications. In D.M. Gabbay and H.J. Ohlbach, editors, *Temporal Logic — Proceedings of the First International Conference*, volume 827 of *Lecture Notes in Artificial Intelligence*, pages 480–505. Springer-Verlag, 1994.

[FO95] M. Fisher and R. Owens. *Executable Modal and Temporal Logics, Proceedings of the IJCAI'93 workshop*, volume 897 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag, 1995.

[FSGW96] D. Fensel, A. Schonegge, R. Groenboom, and B. Wielinga. Specification and verification of knowledge-based systems. In B.R. Gaines and M.A. Musen, editors, *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'96*, pages 4/1–4/20. SRDG Publications, Department of Computer Science, University of Calgary, 1996.

[FW97] M. Fisher and M. Wooldridge. On the formal specification and verification of multi-agent systems. *International Journal of Co-operative Information Systems*, 6(1), 1997. Special issue on Formal Methods in Co-operative Information Systems: Multi-Agent Systems, edited by M. Huhns and M. Singh.

[Gab82] D.M. Gabbay. Intuitionistic basis for non-monotonic logic. In G. Goos and J. Hartmanis, editors, *6th Conference on Automated Deduction*, volume 138 of *Lecture Notes in Computer Science*, pages 260–273. Springer-Verlag, 1982.

[Gab85] D.M. Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. In K.R. Apt, editor, *Logics and Models of Concurrent Systems*, volume F13 of *NATO ASI Series*, pages 439–457. Springer-Verlag, 1985.

[Gab89] D.M. Gabbay. The declarative past and imperative future: Executable temporal logic for interactive systems. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, *Temporal Logic in Specification*, volume 398 of *Lecture Notes in Computer Science*, pages 409–448. Springer-Verlag, 1989.

[Gär92a] P. Gärdenfors. *Belief Revision*, volume 29 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1992.

[Gär92b] P. Gärdenfors. Belief revision: An introduction. In *Belief Revision* [Gär92a], pages 1–28.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability*. W.H. Freeman and Company, New York, 1979.

[GK94] O. Grumberg and R.P. Kurshan. How linear can branching-time be? In D.M. Gabbay and H.J. Ohlbach, editors, *Temporal Logic, Proceedings of the First International Conference on Temporal Logic, ICTL'94*, pages 180–194. Springer-Verlag, 1994.

[GL88] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the International Conference on Logic Programming, ICLP-88*, pages 1070–1080. MIT Press, 1988.

[GL90] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In *Proceedings of the 7th International Conference on Logic Programming, ICLP-90*, pages 579–597. MIT Press, 1990.

[GL91] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[GLPT91] M. Gelfond, V. Lifschitz, H. Przymusinska, and M. Truszczyński. Disjunctive defaults. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, KR'91*, pages 230–237. Morgan Kaufmann, 1991.

[GM94a] P. Gardenfors and D. Makinson. Nonmonotonic inference based on expectations. *Artificial Intelligence*, 65:197–245, 1994.

[GM94b] G. Gottlob and Z. Mingyi. Cumulative default logic: Finite characterization, algorithms, and complexity. *Artificial Intelligence*, 69:329–345, 1994.

[Gol92] R. Goldblatt. *Logics of Time and Computation*, volume 7 of *CSLI Lecture Notes*. Stanford University, 2nd edition, 1992.

[Got92] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, 1992.

[Gro88] A. Grove. Two modelings for theory change. *Journal of Philosophical Logic*, 17:157–170, 1988.

[Gro95] W. Groeneveld. *Logical Investigations into Dynamic Semantics*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 1995.

[GT94] I.S. Gavrila and J. Treur. A formal model for the dynamics of compositional reasoning systems. In A.G. Cohn, editor, *Proceedings of the 11th European Conference on Artificial Intelligence, ECAI'94*, pages 307–311. Wiley and Sons, 1994.

[GTG93] E. Giunchiglia, P. Traverso, and F. Giunchiglia. Multi-context systems as a specification framework for complex reasoning systems. In Treur and Wetter [TW93], pages 45–72.

[Hal97] J. Y. Halpern. A theory of knowledge and ignorance for many agents. *Journal of Logic and Computation*, 7(1):79–108, 1997.

[Hay73] P.J. Hayes. The frame problem and related problems in artificial intelligence. In A. Elithorn and D. Jones, editors, *Artificial and Human Thinking*, pages 45–49. Jossey-Bass, San Francisco, 1973. Also in B.L. Webber and N.J. Nilsson, editors, *Readings in Artificial Intelligence*, Morgan Kaufmann, 1981.

[HC84] G.E. Hughes and M.J. Cresswell. *A Companion to Modal Logic*. Methuen, London, 1984.

[Her94] H. Herre. Compactness properties of nonmonotonic inference operations. In C. MacNish, D. Pearce, and L.M. Pereira, editors, *Logics in Artifical Intelligence, Proceedings JELIA'94*, volume 838 of *Lecture Notes in Artificial Intelligence*, pages 19–33. Springer-Verlag, 1994. Also appeared as "Generalized compactness of nonmonotonic inference operations" in the *Journal of Applied Non-Classical Logics*, 5(1):121–135, 1995.

[HJW97] H. Herre, J. Jaspars, and G. Wagner. Partial logics with two kinds of negation as a foundation for knowledge-based reasoning. In D. Gabbay and H. Wansing, editors, *What is negation?* Oxford University Press, 1997.

[HLMT93] F. van Harmelen, R. Lopez de Mantaras, J. Malec, and J. Treur. Comparing formal specification languages for complex reasoning systems. In Treur and Wetter [TW93], pages 257–282.

[HM85a] J.Y. Halpern and Y. Moses. A guide to the modal logics of knowledge and belief. In A. Joshi, editor, *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 480–490. Morgan Kaufmann, 1985.

[HM85b] J.Y. Halpern and Y. Moses. Towards a theory of knowledge and ignorance: preliminary report. In K.R. Apt, editor, *Logics and Models of Concurrent Systems*, volume F13 of *NATO ASI Series*, pages 459–476. Springer-Verlag, 1985.

[HM90] J.Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990.

[HMT94] W. van der Hoek, J.-J.Ch. Meyer, and J. Treur. Formal semantics of temporal epistemic reflection. In L. Fribourg and F. Turini, editors, *Logic Program Synthesis and Transformation — Meta-Programming in Logic, Proceedings LOPSTR'94 and META'94*, Lecture Notes in Computer Science, pages 332–352. Springer-Verlag, 1994.

[HMT95] W. van der Hoek, J.-J.Ch. Meyer, and J. Treur. Temporalizing epistemic default logic. In R.B. Feenstra and R. Wieringa, editors, *Information Systems - Correctness and Reusability, Selected papers from the IS-CORE-95 Workshop*, pages 173–190. World Scientific Publishers, London, 1995. Extended version appeared in the *Journal of Logic, Language and Information*, 7:341-367, 1998.

[Hod93] W. Hodges. *Model Theory.* Cambridge University Press, 1993.

[Hoo94] J. Hooman. Compositional verification of a distributed real-time arbitration protocol. *Real-Time Systems*, 6:173–206, 1994.

[HV89] J.Y. Halpern and M.Y. Vardi. The complexity of reasoning about knowledge and time I: Lower bounds. *Journal of Computer and System Sciences*, 38:195–237, 1989.

[HW97a] H. Herre and G. Wagner. Semantics for extended generalized logic programs. Technical report, Institut für Informatik, Universität Leipzig, 1997.

[HW97b] H. Herre and G. Wagner. Stable models are generated by a stable chain. *Journal of Logic Programming*, 30(2):165–177, 1997.

[Jas94] J.O.M. Jaspars. *Calculi for Constructive Communication.* Illc dissertation series 1994-4, itk dissertation series 1994-1, Institute for Logic, Language and Computation (ILLC), Amsterdam, 1994.

[Joh90] D.S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. A: Algorithms and complexity*, pages 67–161. Elsevier Science Publishers, Amsterdam, 1990.

[JSHS96] R. Jungclaus, G. Saake, Th. Hartmann, and C. Sernadas. TROLL—a language for object-oriented specification of information systems. *ACM Transactions on Information Systems*, 14(2):175–211, 1996.

[JT97] C.M. Jonker and J. Treur. Compositional verification of multi-agent systems: a formal analysis of pro-activeness and reactiveness. In W.P. De Roever, H. Langmaack, and A. Pnueli, editors, *Proceedings of the International Symposium on Compositionality, COMPOS'97.* Springer-Verlag, 1997. In Press.

[KLM90] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[Kon88a] K. Konolidge. Hierarchic autoepistemic theories for nonmonotonic reasoning. In *Proceedings AAAI'88*, 1988.

[Kon88b] K. Konolige. On the relation between default logic and autoepistemic logic. *Artificial Intelligence*, 35:343–382, 1988.

[Kon94] K. Konolige. Autoepistemic logic. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 217–295. Clarendon Press, Oxford, 1994.

[Kow74] R.A. Kowalski. Predicate logic as a programming language. *Information Processing*, 74:569–574, 1974.

[Kri65] S. Kripke. Semantical analysis of intuitionistic logic. In J.N. Crossley and M. Dummett, editors, *Formal Systems and Recursive Function Theory*, pages 92–129. North Holland, 1965.

[Kro87] F. Kroeger. *Temporal Logic of Programs.* Springer–Verlag, Berlin, 1987.

[Lad77] R.E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput.*, 6:467–480, 1977.

[Lan88] T. Langholm. *Partiality, Truth and Persistence.* Number 15 in CSLI Lecture Notes. Stanford University, 1988.

[Lev84] H.J. Levesque. A logic of implicit and explicit belief. In *Proceedings National Conference on Artificial Intelligence, AAAI-84*, pages 198–202. William Kaufmann, 1984.

[Lev90] H.J. Levesque. All I know: A study in autoepistemic logic. *Artificial Intelligence*, 42:263–309, 1990.

[Lew73] D. K. Lewis. *Counterfactuals*. Blackwell, Oxford, 1973.

[Lif94] V. Lifschitz. Circumscription. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 297–352. Oxford University Press, 1994.

[Lin91] S. Lindström. A semantic approach to nonmonotonic reasoning: inference operations and choice. Technical report, Department of Philosophy, Uppsala University, Sweden, 1991.

[Lin96] B. van Linder. *Modal Logics for Rational Agents*. PhD thesis, Universiteit Utrecht, Faculty of Mathematics and Computer Science, 1996.

[Lin97] F. Lin. Applications of the situation calculus to formalizing control and strategic information: the Prolog cut operator. In M.E. Pollack, editor, *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pages 1412–1418. Morgan Kaufmann, 1997.

[Llo87] J. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 1987. Second, extended edition.

[Lou87] R. Loui. Defeat among arguments: A system of defeasible inference. *Computational Intelligence*, 3(3):100–107, 1987.

[LPT92] I. van Langevelde, A. Philipsen, and J. Treur. Formal specification of compositional architectures. In B. Neumann, editor, *Proceedings of the 10th European Conference on Artificial Intelligence, ECAI'92*, pages 272–276. John Wiley & Sons, 1992.

[LR96] F. Lin and R. Reiter. Rules as actions: A situation calculus semantics for logic programs. *Journal of Logic Programming*, 31:299–330, 1996.

[LS92] F. Lin and Y. Shoham. A logic of knowledge and justified assumptions. *Artificial Intelligence*, 57:271–289, 1992.

[LT89] P.H.G. van Langen and J. Treur. Representing world situations and information states by many-sorted partial models. Technical Report PE8904, University of Amsterdam, Department of Mathematics and Computer Science, 1989.

[Łuk90] W. Łukaszewicz. *Non-monotonic Reasoning: Formalization of Commonsense Reasoning.* Ellis Horwood, New York, 1990.

[Mak89] D. Makinson. General theory of cumulative inference. In M. Reinfrank, J. de Kleer, M.L. Ginsberg, and E. Sandewall, editors, *Non-Monotonic Reasoning, Proceedings of the Second International Workshop on Non-Monotonic Reasoning*, volume 346 of *Lecture Notes in Artificial Intelligence*, pages 1–18. Springer-Verlag, 1989.

[Mak94] D. Makinson. General patterns in nonmonotonic reasoning. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 35–110. Clarendon Press, Oxford, 1994.

[McC77] J. McCarthy. Epistemological problems of artificial intelligence. In R. Reddy, editor, *Proceedings 5th International Joint Conference on Artificial Intelligence*, pages 1038–1044. Morgan Kaufmann, Los Altos, CA, 1977.

[McC80] J. McCarthy. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39, 1980.

[MD80] D. McDermott and J. Doyle. Nonmonotonic logic I. *Artificial Intelligence*, 13(1-2):41–72, 1980.

[MD82] D. McDermott and J. Doyle. Nonmonotonic logic II. *Journal of the ACM*, 29(1):33–57, 1982.

[MH69] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer, D. Michie, and M. Swann, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[MH95] J.-J. Ch. Meyer and W. van der Hoek. *Epistemic Logic for Computer Science and Artificial Intelligence*, volume 41 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1995.

[MN88] P. Maes and D. Nardi, editors. *Meta-level architectures and reflection.* Elsevier Science Publishers, 1988.

[Moo85] R.C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems, Vol. 1: Specification.* Springer-Verlag, 1992.

[MST93] V.W. Marek, G.F. Schwarz, and M. Truszczyński. Modal nonmonotonic logics: Ranges, characterization, computation. *Journal of the ACM*, 40:963–990, 1993.

[MT89] V.W. Marek and M. Truszczyński. Relating autoepistemic and default logics. In R.J. Brachman, H.J. Levesque, and R. Reiter, editors, *Proceedings of the First International Conference on the Principles of Knowledge Representation and Reasoning*, pages 276–288. Morgan Kaufmann, 1989.

[MT92] V.W. Marek and M. Truszczyński. More on modal aspects of default logic. *Fundamenta Informaticae*, 17:99–116, 1992.

[MT93] V.W. Marek and M. Truszczyński. *Nonmonotonic Logic: Context-dependent Reasoning.* Springer-Verlag, 1993.

[MTT97] V.W. Marek, J. Treur, and M. Truszczyński. Representation theory for default logic. *Annals of Mathematics and Artificial Intelligence*, 21:343–358, 1997.

[Nie95] I. Niemelä. Towards efficient default reasoning. In *Proceedings 14th International Joint Conference on Artificial Intelligence*, pages 312–318. Morgan Kaufmann, 1995.

[Nie96] I. Niemelä. Implementing circumscription using a tableau method. In W. Wahlster, editor, *Proceedings 12th European Conference on Artificial Intelligence, ECAI'96*, pages 80–84. John Wiley & Sons, 1996.

[NKP94] M. Nirkhe, S. Kraus, and D. Perlis. Thinking takes time: A modal active-logic for reasoning IN time. Technical Report CS-TR-3249, UMIACS-TR-94-39, University of Maryland, 1994.

[Nut84] D. Nute. Conditional logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, Vol. II*, pages 388–439. Reidel, Dordrecht, 1984.

[OM44] M.A. Orgun and W. Ma. An overview of temporal and modal logic programming. In D.M. Gabbay and H.J. Ohlbach, editors, *Temporal Logic, Proceedings ICTL'94*, volume 827 of *Lecture Notes in Artificial Intelligence*, pages 445–479. Springer-Verlag, 1944.

[Pol87] J. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.

[Poo88] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[Poo89] D. Poole. What the lottery paradox tells us about default reasoning. In R.J. Brachman, H.J. Levesque, and R. Reiter, editors, *Proceedings First International Conference on Principles of Knowledge Representation and Reasoning*, pages 333–340. Morgan Kaufmann, 1989.

[Pop77] K. Popper. *The Logic of Scientific Discovery.* Hutchinson, London, 1977. 9th edition.

[Pra97] H. Prakken. *Logical Tools for Modelling Legal Argument.* Kluwer Academic Publishers, Dordrecht, 1997.

[Prz90] T. Przymusinski. Well-founded semantics coincides with three-valued stable semantics. *Fundamenta Informaticae*, 13:445–463, 1990.

[PS92] C.H. Papadimitriou and M. Sideri. On finding extensions of default theories. In J. Biskup and R. Hull, editors, *Proceedings 4th International Conference on Database Theory, ICDT-92*, volume 646 of *Lecture Notes in Computer Science*, pages 276–281. Springer-Verlag, 1992.

[PS96] H. Prakken and G. Sartor. A system for defeasible argumentation, with defeasible priorities. In D.M. Gabbay and H.J. Ohlbach, editors, *Practical reasoning, Proceedings FAPR'96*, volume 1085 of *Lecture Notes in Artificial Intelligence*, pages 510–524. Springer-Verlag, 1996.

[Rab77] M. O. Rabin. Decidable theories. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 595–629. North-Holland, Amsterdam, 1977.

[Rei80a] R. Reiter. Equality and domain closure in first-order data bases. *Journal of the ACM*, 27:235–249, 1980.

[Rei80b] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[Rei91] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 418–420. Academic Press, 1991.

[RG91] A. S. Rao and M. P. Georgeff. Modeling rational agents withing a BDI-architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of the Second Conference on Knowledge Representation and Reasoning, KR'91*, pages 473–484. Morgan Kaufmann, 1991.

[RG92] A.S. Rao and M.P. Georgeff. An abstract architecture for rational agents. In C. Rich, W. Swartout, and B. Nebel, editors, *Proceedings of the third International Conference on Principles of Knowledge Representation and Reasoning, KR'92*, pages 439–449. Morgan Kaufmann, 1992.

[Rij93] M. de Rijke. *Extending Modal Logic.* Illc dissertation series 1993-4, Institute for Logic, Language and Computation (ILLC), Amsterdam, 1993.

[RS94] V. Risch and C.B. Schwind. Tableau-based characterization and theorem proving for default logic. *Journal of Automated Reasoning*, 13:223–242, 1994.

[San85] E. Sandewall. A functional approach to non-monotonic logics. *Computational Intelligence*, 1:80–87, 1985.

[SC85] A.P. Sistla and E.M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

[Sch91] T. Schaub. Assertional default theories: A semantical view. In J.A. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning*, pages 496–506. Morgan Kaufmann, 1991.

[Sch92a] T. Schaub. On constrained default theories. In B. Neumann, editor, *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 304–308. John Wiley & Sons, 1992.

[Sch92b] K. Schlechta. Some results on classical preferential models. *Journal of Logic and Computation*, 2(6):675–686, 1992.

[Sch95a] K. Schlechta. Preferential choice representation theorems for branching time structures. *Journal of Logic and Computation*, 5(6):783–800, 1995.

[Sch95b] G. Schwarz. In search of a "true" logic of knowledge: the nonmonotonic perspective. *Artificial Intelligence*, 79:39–63, 1995.

[Sho87] Y. Shoham. Nonmonotonic logics: Meaning and utility. In J. McDermott, editor, *Proceedings 10th International Joint Conference on Artificial Intelligence*, pages 388–392. Morgan Kaufmann, 1987.

[Sho88] Y. Shoham. *Reasoning about Change*. MIT Press, Cambridge, 1988.

[Sin94] M. P. Singh. *Multiagent Systems: A Theoretical Framework for Intentions, Know-how, and Communications*, volume 799 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1994.

[Spa90] E. Spaan. Nexttime is not necessary (extended abstract). In R. Parikh, editor, *Theoretical Aspects of Reasoning About Knowledge, Proceedings of the Third Conference (TARK 1990)*, pages 241–256. Morgan Kaufmann, 1990.

[SS95] E. Sandewall and Y. Shoham. Non-monotonic temporal reasoning. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 4: Epistemic and Temporal Reasoning*, pages 439–498. Clarendon Press, Exford, 1995.

[ST94] G. Schwarz and M. Truszczyński. Minimal knowledge problem: a new approach. *Artificial Intelligence*, 67:113–141, 1994.

[Sta68] R. Stalnaker. A theory of conditionals. In N. Rescher, editor, *Studies in Logical Theory*, number 2 in American Philosophical Quarterly Monograph Series. Blackwell, Oxford, 1968.

[Sta93] R. Stalnaker. A note on non-monotonic modal logic. *Artificial Intelligence*, 64(2):183–196, 1993.

[Ste95] M. Stefik. *Introduction to Knowledge Systems.* Morgan Kaufmann, 1995.

[Sti92] J. Stillman. The complexity of propositional default logics. In W. Swartout, editor, *Proceedings AAAI-92*, pages 794–799. AAAI Press, Menlo Park, CA, 1992.

[Tar56] A. Tarski. *Logic, Semantics, Metamathematics, Papers from 1923–1938.* Clarendon Press, 1956.

[Thi92] E. Thijsse. *Partial logic and knowledge representation.* PhD thesis, ITK, Tilburg University, 1992.

[Tre91] J. Treur. Declarative functionality descriptions of interactive reasoning modules. In H. Boley and M.M. Richter, editors, *Processing Declarative Knowledge, Proceedings of the International Workshop, PDK-91*, volume 567 of *Lecture Notes in Artificial Intelligence*, pages 221–236. Springer-Verlag, 1991.

[Tre94] J. Treur. Temporal semantics of meta-level architectures for dynamic control of reasoning. In L. Fribourg and F. Turini, editors, *Logic Program Synthesis and Transformation — Meta-Programming in Logic: Proceedings LOPSTR'94 and META'94*, volume 883 of *Lecture Notes in Computer Science*, pages 353–376. Springer-Verlag, 1994.

[TT91] Y.-H. Tan and J. Treur. A bi-modular approach to nonmonotonic reasoning. In M. De Glas and D. Gabbay, editors, *Proceedings World Congress on Fundamentals of Artificial Intelligence, WOCFAI-91*, pages 461–476, 1991.

[TT92] Y.-H. Tan and J. Treur. Constructive default logic and the control of defeasible reasoning. In B. Neumann, editor, *Proceedings of the 10th European Conference on Artificial Intelligence, ECAI'92*, pages 299–303. John Wiley & Sons, 1992.

[TW93] J. Treur and Th. Wetter, editors. *Formal Specification of Complex Reasoning Systems.* Ellis Horwood, 1993.

[TW94] J. Treur and M. Willems. A logical foundation for verification. In A.G. Cohn, editor, *Proceedings of the Eleventh European Conference on Artificial Intelligence, ECAI'94*, pages 745–749. John Wiley & Sons, 1994.

[VBJR95] A. Visser, J. van Benthem, D. de Jongh, and G.R. Renardel de Lavalette. NNIL, a study in intuitionistic propositional logic. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra*, volume 53 of *CSLI Lecture Notes*, pages 289–326. Stanford University, 1995.

[Vel85] F. Veltman. *Logics for Conditionals.* PhD thesis, Jurriaans, Amsterdam, 1985.

[Vel96] F. Veltman. Defaults in update semantics. *Journal of Philosophical Logic*, 25:221–261, 1996.

[Voo93] F. Voorbraak. Preference-based semantics for nonmonotonic logics. In R. Bajcsy, editor, *Proceedings 13th International Joint Conference on Artificial Intelligence*, pages 584–589. Morgan Kaufmann, 1993.

[Vre92] G. Vreeswijk. Nonmonotonicity and partiality in defeasible argumentation. In W. van der Hoek, J.-J. Ch. Meyer, Y.-H. Tan, and C. Witteveen, editors, *Non-monotonic Reasoning and Partial Semantics*, pages 157–180. Ellis Horwood, 1992.

[Wal93] M.G. Wallace. Tight, consistent, and computable completions for unrestricted logic programs. *Journal of Logic Programming*, 15:243–273, 1993.

[War72] L.E. Ward. *Topology.* Marcel Dekker, Inc., New York, 1972.

[Wey80] R.W. Weyhrauch. Prolegomena to a theory of mechanized formal reasoning. *Artificial Intelligence*, 13:133–170, 1980.

[WJ95] M.J. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[ZR97] G.Q. Zhang and W.C. Rounds. Nonmonotonic consequences of default domain theory. *Annals of Mathematics and Artificial Intelligence*, 20:227–265, 1997.

Samenvatting

De dynamiek van redeneren

In de Kunstmatige Intelligentie probeert men systemen te ontwerpen die intelligent gedrag vertonen. Redeneren is een vorm van intelligent gedrag en is iets wat men ook een computer wil kunnen laten doen. Hiertoe moet eerst geanalyseerd worden hoe dat redeneren werkt. Sinds het begin van de jaren tachtig zijn er veel formalisaties van menselijk redeneergedrag voorgesteld, maar geen van deze formalisaties bleek in elke situatie te voldoen. Daarop bedacht men dat het zinnig zou zijn om naar de abstracte eigenschappen van redeneren te kijken, opdat vastgesteld zou kunnen worden aan welke eisen een formalisatie moet voldoen, afhankelijk van de situatie. Op het meest abstracte niveau waarop naar redeneren gekeken wordt, beschouwt men redeneren als een invoer-uitvoer proces: gegeven bepaalde beginfeiten en beginkennis, levert een redenerend systeem (kunstmatig of natuurlijk) een verzameling conclusies. Hier wordt dus alleen naar het eindprodukt van het redeneerproces gekeken en niet naar hoe dat eindprodukt tot stand gekomen is.

De belangrijkste bijdrage van dit proefschrift is dat er een raamwerk opgezet wordt waarbinnen ook de dynamische aspecten van redeneren op voldoende abstract niveau beschreven en bestudeerd kunnen worden. In het bijzonder worden twee elementen van redeneerprocessen bekeken. In de eerste plaats is dat het niet-deterministische karakter van redeneren. Vaak moeten er tijdens het redeneren keuzes gemaakt worden, bijvoorbeeld als er aanvullende aannames gemaakt moeten worden, of als er conflicterende kennis is (welke kennis moet er dan verworpen worden?). Op dit abstractieniveau kan men redeneren dus zien als een proces waarbij bij een gegeven verzameling beginfeiten en beginkennis meerdere mogelijke conclusieverzamelingen gegenereerd worden. Ten tweede kan men kijken naar het proces van redeneren: een redenerend systeem begint met de beginfeiten, waarop het bijvoorbeeld redeneerregels kan toepassen om tot extra conclusies te komen. In deze toestand kan het systeem weer nieuwe regels op de nieuwe conclusies toepassen. Of er kunnen extra aannames of observaties gedaan worden. Gedurende het proces van redeneren (dat niet per se hoeft te eindigen) passeert het systeem meerdere interne *informatietoestanden* (die bepalen wat het systeem weet op dat moment). Op dit abstractieniveau kan men kijken naar de opeenvolgingen van informatietoestanden van het systeem tijdens het redeneren, daarbij nog steeds abstraherend van de implementatie (fysiek of als software) van het redenerende systeem.

In Hoofdstuk 1 worden 4 verschillende abstractieniveaus om redeneren op te beschrijven, onderscheiden. Op het eerste niveau wordt redeneren beschouwd als invoer-uitvoer functie, op het tweede niveau als functie die bij gegeven invoer meerdere mogelijke conclusieverzamelingen geeft. Op het derde niveau wordt redeneren gezien als een verzameling informatietoestand-sequenties, *traces* genoemd, die beginnen in een toestand waarin alleen beginfeiten gekend worden. Op het vierde niveau is redeneren beschreven door het specificeren van een redenerend systeem (dat kan een implementatie in een programmeertaal zijn, een blauwdruk van een fysieke machine, of zelfs een beschrijving van een mens).

De precieze wiskundige beschrijving van deze niveaus (de formalisatie) is het onderwerp van Hoofdstuk 2. Hoe specificeer je nu redeneerpatronen op deze abstractieniveaus? Dat kan natuurlijk in algemene wiskundige taal, maar nog beter is het als je een formele taal hebt die speciaal geschikt is voor dit doel. Als er voor deze taal bovendien nog een bewijssysteem bestaat, is het mogelijk om formele bewijzen over redeneergedrag te maken. Dit soort talen noemen we in dit proefschrift *specificatietalen*, en in Hoofdstuk 3 passeren enkele mogelijke specificatietalen de revue.

Het grootste deel van dit proefschrift gaat vervolgens over het derde niveau, met uitzondering van Sectie 8.2 en Hoofdstuk 10. Dat laatste hoofdstuk geeft een eerste analyse van de formalisatie van niveau 2, met mogelijke eigenschappen van redeneerpatronen op dat niveau, onderliggende semantieken en relaties met eigenschappen op het eerste niveau bekend uit de literatuur. In Sectie 8.2 wordt de formalisatie op niveau 2 gebruikt voor de beschrijving van een redeneertaak uit de praktijk, te weten ecologische classificatie van stukken grond op basis van de erop aangetroffen plantensoorten.

Redeneren is een proces dat plaatsvindt in de tijd, en de stapjes tussen opeenvolgende informatietoestanden zijn stapjes in de tijd. Deze observatie leidt tot een klasse natuurlijke specificatietalen voor redeneergedrag, namelijk die van temporele informatielogica's. In zo'n formele taal kan de veranderende informatie door de tijd heen beschreven worden. De precieze invulling van de informatietoestanden en de taal om deze te beschrijven en de precieze invulling van de tijd leiden tot meerdere mogelijke logica's. In Hoofdstuk 4 worden een aantal hiervan beschreven, gebruikmakend van partiële logica en epistemische logica voor de informatietoestanden, en gebruikmakend van lineaire en vertakkende discrete tijd. Tevens wordt een aantal manieren beschreven waarop met behulp van een ordening bepaalde temporele informatielogica-modellen geselecteerd kunnen worden (de *beoogde* of 'intended' modellen).

De geschiktheid van deze logica's om redeneergedrag mee te beschrijven wordt in Hoofdstuk 5 aangetoond door bestaande formalisaties van verschillende vormen van redeneergedrag in deze logica's te beschrijven. Dit wordt gedaan voor default-logica (zowel op basis van lineaire als van vertakkende tijd), logisch programmeren, klassieke bewijsvoering, autoepistemische logica en de logica GK. Dit laat zien dat deze redeneervormen een natuurlijke beschrijving hebben op het derde abstractieniveau, en dat temporele informatielogica's krachtig genoeg zijn om deze redeneervormen op natuurlijke wijze te beschrijven.

Hoofdstuk 6 gaat in op executie van temporele theorieën. Het idee is hier dat als je een algemeen executiemechanisme hebt voor temporele informatielogica-theorieën, die vormen van redeneergedrag kunnen beschrijven, dan heb je een algemeen redeneermechanisme. Dit 'executeren' van theorieën betekent eigenlijk het vinden van een model voor zo'n theorie, op gestructureerde wijze (gebruikmakend van heuristische zoekkennis). Naast een executiealgorithme voor een bepaalde klasse temporele theorieën, wordt ook een compositionele architectuur voor een systeem beschreven, dat dit algorithme kan uitvoeren.

Naast de geschiktheid van een taal om dingen in uit te drukken in meer informele zin, kan men ook kijken naar de wiskundige expressiviteit. De vraag is dan: wat kan je precies in principe wel, en wat niet uitdrukken in een bepaalde taal? In Hoofdstuk 7 wordt gekeken naar de wiskundige expressiviteit van een bepaalde klasse theorieën in één van de temporele informatielogica's. Ook wordt er gekeken naar de expressiviteit van (infinitaire) default-logica als specificatietaal voor redeneren op het tweede en derde abstractieniveau.

Hoofdstuk 8 bevat, naast de analyse van ecologische classificatie, nog een andere toepassing van de ontwikkelde theorie. Dit betreft de mogelijkheid om in een temporele informatielogica compositionele multi-agentsystemen te beschrijven. Dit zijn systemen waarin er meerdere onafhankelijke redenerende agenten zijn. Om dit soort systemen te beschrijven wordt een temporele informatielogica uitgebreid met de mogelijkheid om er verschillende informatietoestanden op hetzelfde tijdstip (behorend bij meerdere agenten) in te beschrijven. Deze logica kan gebruikt worden om bewijzen van eigenschappen van dit soort systemen mee te formaliseren.

In Hoofdstuk 9 komen een aantal logische thema's aan de orde. Voor één van de temporele informatielogica's wordt een bewijssysteem beschreven. Tevens wordt van deze logica, en van een erop gebaseerde, MTEL, de beslisbaarheid bewezen en de precieze complexiteit. Hoewel deze laatste logica niet-monotoon is, hetgeen wil zeggen dat conclusies uit een premisse getrokken hun geldigheid kunnen verliezen als de premisse sterker wordt, blijft monotonie behouden voor een bepaalde klasse formules. Dit fenomeen geldt algemener voor zogenaamde preferentiële logica's, en dit is het onderwerp van de tweede sectie van Hoofdstuk 9. Ook is MTEL niet *cumulatief*. Het laatste gedeelte van dit hoofdstuk geeft semantische representatieresultaten voor niet-cumulatieve afleidingssystemen.

Na het eerder genoemde Hoofdstuk 10, geeft Hoofdstuk 11 globale conclusies en suggesties voor verder onderzoek. De belangrijkste conclusie is dat het mogelijk, interessant, nuttig en soms noodzakelijk is om complexe redeneervormen te analyseren op een abstractieniveau tussen niveaus 1 en 4. In het bijzonder kunnen de niet-deterministische en dynamische aspecten van redeneren gemodelleerd en bestudeerd worden en dit heeft een sterke toegevoegde waarde.